SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

Tecnológico Nacional de México
Coordinación Sectorial Académica

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

TESIS DE MAESTRÍA EN CIENCIAS

Incremento de la Eficiencia del Algoritmo K-means Mediante la Mejora de la Heurística Early Classification

presentada por

Ing. Vitervo López Caballero

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Director de tesis
Dr. Joaquín Pérez Ortega

Codirectora de tesis
Dra. Adriana Mexicano Santoyo

Cuernavaca, Morelos, México. Marzo de 2015.

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO
Centro Nacional de Investigación y Desarrollo Tecnológico

"2015, Año del Generalísimo José María Morelos y Pavón"

Cuernavaca, Mor., 16/febrero/2015
OFICIO No. DCC/018/2015

Asunto: Aceptación de documento de tesis

C. DR. GERARDO V. GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **C. Vitervo López Caballero**, con número de control M13CE029, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado **"Incremento de la eficiencia del algoritmo K-Means mediante la mejora de la heurística Early Classification"** 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

CO-DIRECTORA DE TESIS

Dra. Adriana Mexicano Santoyo
Doctora en Ciencias de la Computación
8041310

REVISOR 1

Dr. José Crispín Zavala Díaz
Doctor en Ciencias Computacionales
3406871

REVISOR 2

M.C. Humberto Hernández García
Maestro en Ciencias con Especialidad
en Sistemas Computacionales
7573641

REVISOR 3

Dr. Juan Carlos Rojas Pérez
Doctor en Ciencias en Ciencias de la Computación
6099372

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

AMR/lmz

"2015, Año del Generalísimo José María Morelos y Pavón"

Cuernavaca, Mor., 20 de febrero de 2015
OFICIO No. SAC/069/2015

Asunto: Autorización de impresión de tesis

ING. VITERVO LÓPEZ CABALLERO
CANDIDATO AL GRADO DE MAESTRO 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 **"Incremento de la eficiencia del algoritmo K-Means mediante la mejora de la heurística Early Classification"**, 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.

ATENTAMENTE
"CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO

CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SUBDIRECCIÓN ACADEMICA

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

GVGR/mcr

DEDICATORIA

A Dios, por dejarme cumplir este sueño tan anhelado y guiar mi vida por buenos pasos, a pesar de los obstáculos.

A mi padre Pedro López García† porque sé que donde quiera que estés me estás cuidando y guiando por los pasos correctos.

A mi madre Juana Caballero Vásquez, por su inagotable amor, sus consejos, por apoyarme siempre en mis decisiones y enseñarme a ser una persona humilde.

A mi hermano Austreberto, por ser como mi segundo padre, por su gran apoyo y sus valiosos consejos.

A toda mi familia, por todo el apoyo brindado durante esta etapa de mi formación profesional y por confiar en mí.

AGRADECIMIENTOS

Mi más profundo agradecimiento a mi amigo y director de tesis, el Dr. Joaquín Pérez Ortega, por guiarme generosamente en esta etapa de mi formación profesional y por su valiosa y oportuna opinión científica.

A mi Co-directora Dra. Adriana Mexicano Santoyo, por sus consejos y sus valiosas opiniones.

A mi amigo Dr. José Crispín Zavala Díaz, por ser conmigo una gran persona y por brindarme sus consejos.

A la Dra. Leticia Sánchez Lima, por brindarme su valioso tiempo en la revisión del documento.

A los profesores que formaron parte de mi comité revisor: Dr. Juan Carlos Rojas Pérez, M.C. Humberto Hernández García y Dr. José Crispín Zavala Díaz, por su valioso aporte al desarrollo y conclusión de esta tesis y por su opinión científica.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por brindarme el apoyo económico durante mis estudios de maestría.

Al Centro Nacional de Investigación y Desarrollo Tecnológico por darme la oportunidad de realizar mis estudios y al personal que labora en él, por todas sus atenciones.

A los estudiantes de doctorado: Miguel y Nelva, por apoyarme en el cumplimiento de esta meta.

A los compañeros integrantes de mi equipo de básquetbol, por brindarme su amistad y por los logros obtenidos durante mi estancia en CENIDET.

A mis amigos Ramiro, Omar, Bernardo, Marcos, Alma, Miguel y Ámbar; por apoyarme en las dificultades y alegrías.

RESUMEN

El algoritmo de agrupamiento K-means se ha aplicado en varios dominios debido a su facilidad de implementación computacional. Sin embargo, una de sus limitaciones es su alta complejidad computacional. Por esta razón, en esta investigación se propuso una nueva meta-heurística a la que se le denominó N-means, la cual permite reducir la complejidad de K-means de manera importante.

Como resultado de observar las ejecuciones del algoritmo K-means se determinó que algunos grupos permanecen constantes porque ya no intercambian objetos con otros grupos. Con base en este conocimiento se desarrolló una nueva heurística a la que se denominó *grupos estables*. En dicha heurística se descartan los objetos asignados a un grupo estable en las iteraciones posteriores. En la meta-heurística N-means que se desarrolló en esta investigación se integran las heurísticas *Early Classification* y *grupos estables*.

Para validar N-means, se realizaron experimentos con instancias reconocidas por la comunidad científica. Se reportan los resultados promedio de 30 ejecuciones de cada instancia variando los parámetros de k=100, 200, 400 y 800. Se contrastaron los resultados de los algoritmos K-means, *Early Classification* y N-means.

En esta investigación se obtuvieron importantes resultados, entre los cuales se destacan los siguientes: a) para una instancia sintética de 40,000 objetos, N-means redujo el tiempo de ejecución en 75.5% y una disminución de la calidad de agrupamiento a -1.52%, *Early Classification* redujo el tiempo de ejecución en 67.7% y una disminución de la calidad de agrupamiento a -1.31%; b) con una instancia real de 245,057 objetos, N-means redujo el tiempo de ejecución en 64% y una disminución de la calidad a -4.56%, *Early Classification* redujo el tiempo de ejecución en 49% y una disminución de la calidad a -4.11%.

Finalmente se considera que las mejoras al algoritmo K-means posibilitarán la solución de instancias grandes como las que emergen en el paradigma Big Data.

TABLA DE CONTENIDO

Pág.

LISTA DE FIGURAS

Pág.

LISTA DE TABLAS

Pág.

Capítulo 1
Introducción

"*Para buscar la verdad es necesario ser independiente, completamente independiente*"

H. Poincaré (1854 - 1912)

K-means [1] es uno de los algoritmos de agrupamiento más antiguos. Su investigación se remonta a mediados del siglo pasado [2] y está clasificado como el segundo entre los 10 algoritmos de minería de datos más utilizados [3]. Dicho algoritmo se aplica en diversas áreas del conocimiento, por ejemplo: la epidemiología [4], investigación de mercado, análisis de documentos, análisis financiero, reconocimiento facial [5], reconocimiento de patrones, cuantificación vectorial, detección de anomalías [6], entre otros.

Las principales etapas del algoritmo K-means son las siguientes: inicialización, clasificación, cálculo de centroides y convergencia [7].

En el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) se han venido desarrollando heurísticas para mejorar el algoritmo K-means. Una

de ellas es la heurística *Early Classification* (EC) [7] la cual mejora la eficiencia del algoritmo K-means en la etapa de clasificación.

La presente investigación tiene como objetivo incrementar la eficiencia del algoritmo K-means mediante una mejora a la heurística EC. En particular, en esta investigación se propone una nueva meta-heurística a la que se le denominó N-means, que reduce de manera importante la complejidad del algoritmo K-means.

La tesis está organizada de la siguiente manera: en el Capítulo 2, se realiza un estudio del estado del arte relacionado con este problema de investigación. En el Capítulo 3, se desarrolla la propuesta de la nueva meta-heurística N-means. En el Capítulo 4, se analiza y valida experimentalmente esta meta-heurística N-means. Finalmente, en el Capítulo 5, se exponen las conclusiones y se proponen temas para futuras investigaciones.

1.1. Contexto de la investigación

En el CENIDET, se han realizado investigaciones aplicando el algoritmo K-means como una técnica de agrupamiento, por ejemplo: Barrón [8], Boone [9] y Mexicano [10]. En otras investigaciones se proponen mejoras al algoritmo, como se expone a continuación:

a) Basave [11] presenta una heurística que establece condiciones de convergencia para el algoritmo. Esto se realiza cuando en dos iteraciones sucesivas el valor del error al cuadrado de la última iteración es mayor que el valor del error al cuadrado de la iteración anterior. También cuando los centroides en dos iteraciones sucesivas no cambian.
b) Moreno [12] desarrolla una heurística que consiste en que un objeto sólo puede cambiar de membresía a un grupo vecino adyacente, entre una iteración y otra. Para ello, para cada objeto se realizan ocho cálculos de distancia a sus centroides vecinos.
c) Por otra parte, Pérez [7] presenta una heurística que identifica los objetos con baja probabilidad de cambio de grupo y los excluye de futuros cálculos de distancia.

d) En el artículo propuesto por Pérez [13] se presenta una heurística que consiste en que un objeto sólo puede cambiar de membresía a un grupo vecino adyacente entre una iteración y otra.

Entre los artículos sugeridos, el que tiene mayor relación con la presente investigación es el que se muestra en el inciso “c”. Los artículos descritos en los incisos “c” y “d” se explicarán con mayor detalle en el Capítulo 2.

1.2. Descripción del problema de investigación

Una de las limitaciones del algoritmo K-means es su alto costo computacional [14]. En particular su complejidad se define como $O(nkdt)$ [15], donde n es el número de objetos a agrupar, k es el número de grupos, d son las dimensiones y t es el número de iteraciones. Dicha complejidad se hace notar cuando cualquiera de los parámetros tiende a ser mayor, lo que incide en el tiempo de respuesta del algoritmo.

A manera de ejemplo, en las Figuras (1.1, 1.2, 1.3, que se localizan en las págs. 4 y 5) se muestran en secuencia, los resultados de tres experimentos. Como se aprecia, a medida que aumenta el número de objetos, de grupos y de dimensiones, se incrementa el tiempo de ejecución del algoritmo K-means de manera no lineal. A continuación, se describen detalladamente los tres experimentos.

a) Para el primer experimento se incrementa el parámetro n con intervalos de 10,000 en 10,000 hasta 100,000. Se establecen valores para los parámetros de $k = 16$ y $d = 2$ en todos los casos. En este experimento se usó una instancia de 100,000 objetos con distribución uniforme. (Ver Figura 1.1 que se localiza en la pág. 4)
b) Para el segundo experimento se incrementa el parámetro k con intervalos de 2 en 2 hasta 192. Se establecen valores para los parámetros de $n = 100{,}000$ y $d = 2$ en todos los casos. En este experimento se usó una instancia de 100,000 objetos con distribución uniforme. (Ver Figura 1.2 que se localiza en la pág. 4)

c) Finalmente, en el tercer experimento se incrementa el parámetro d con intervalos de 32 en 32 hasta 1024. Se establecen valores para los parámetros de $n = 1024$ y $k = 16$. En este caso particular se usaron seis bases de datos de 1024 objetos con distribución uniforme. (Ver Figura 1.3 que se localiza en la pág. 5)

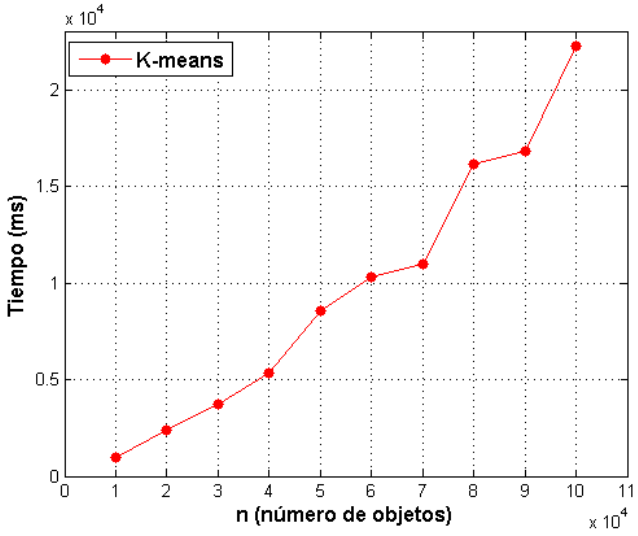

Figura 1.1. Tendencia al incrementar el parámetro n (número de objetos).

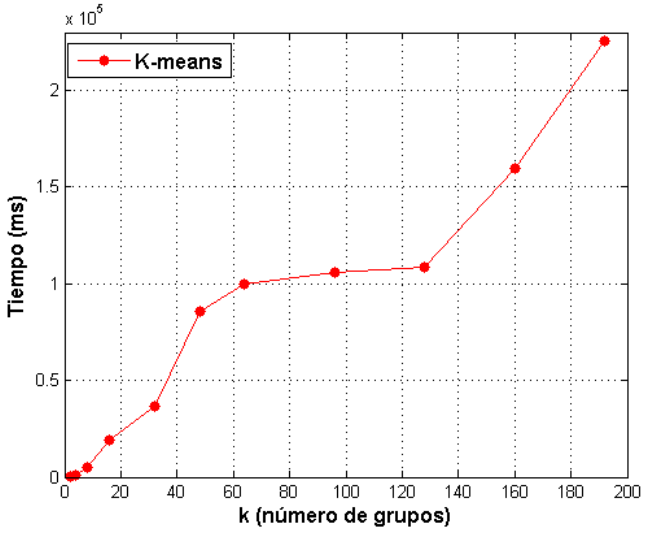

Figura 1.2. Tendencia al incrementar el parámetro k (número de grupos).

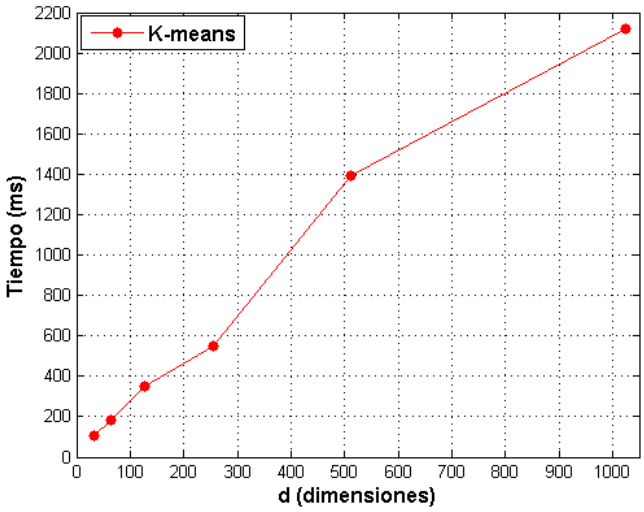

Figura 1.3. Tendencia al incrementar el parámetro d (dimensiones).

En el CENIDET se han realizado varias mejoras a este algoritmo. Una de las que ha tenido mayor éxito es la heurística EC. Sin embargo, como resultado de experimentaciones, se encontraron indicios de que se puede mejorar. Por lo anterior el problema que da origen a la presente investigación se focaliza en incrementar la eficiencia de la heurística EC.

1.3. Hipótesis de investigación

Con base en un estudio y un análisis experimental de la heurística EC, es factible proponer una mejora en la eficiencia de dicha heurística sin reducir significativamente la calidad de la solución.

1.4. Justificación

A través de diversas investigaciones se realizaron mejoras al algoritmo K-means. Una de las heurísticas más recientes y alentadoras es EC. Entre los beneficios que aporta una mejora a dicha heurística se encuentran los siguientes:

a) Reduciría la complejidad del algoritmo K-means, lo cual tendría un impacto social ya que es un algoritmo muy utilizado en diferentes áreas del

conocimiento como la epidemiología, marketing, economía, medicina, entre otros.

b) Posibilitaría la solución de instancias mayores como las que emergen en el paradigma Big Data.
c) Será de utilidad en problemas específicos en los cuales se requiera una solución en un tiempo menor al que toma el algoritmo K-means, en los que no sea significativa una pequeña disminución de la calidad.

Con base en los beneficios descritos en los incisos a, b y c, la mejora a la heurística EC ayudará a la solución de problemas de diversas áreas del conocimiento.

1.5. Objetivo de la tesis

Incrementar la eficiencia del algoritmo K-means mediante una mejora de la heurística *Early Classification*.

1.6. Alcances y limitaciones

Alcances

1) La mejora sólo estuvo orientada a la heurística EC.
2) La validación de los resultados se realizó de manera experimental.
3) Se obtuvo una versión mejorada de la heurística EC implementada computacionalmente.

Limitaciones

1) La mejora sólo se aplicó a la heurística EC en la etapa de clasificación del algoritmo K-means.
2) La mejora se validó de manera experimental.
3) La validación experimental de la mejora se desarrolló con datos aceptados por la comunidad internacional especializada en el área y con datos sintéticos.
4) La tesis se desarrolló con el equipo de trabajo disponible en el CENIDET.

Capítulo 2
Revisión del estado del arte

"*El propósito de la investigación, en términos muy generales, es agregar algo nuevo a los conocimientos humanos*"

Santiago Zorrilla Arena (2012)

En este capítulo se describen los artículos de investigación que hacen referencia a los orígenes y mejoras del algoritmo K-means. La descripción se organiza de la siguiente manera: en la Sección 2.1, se hace un estudio de los orígenes del algoritmo. En la Sección 2.2, se describen algunos artículos que presentan mejoras en la fase de inicialización. En la Sección 2.3, se describe un artículo que expone una mejora en la fase de convergencia. Finalmente, en la Sección 2.4, se detallan algunos artículos que presentan mejoras en la fase de clasificación. Cabe destacar que esta investigación se centra en esta última fase, por lo que se profundiza en ella.

2.1. Orígenes del algoritmo K-means

K-means es uno de los algoritmos de agrupamiento más antiguos. Su investigación es realizada por numerosos investigadores en diferentes disciplinas y se remonta a mediados del siglo pasado [2]. Entre los investigadores más notables se encuentran MacQueen [1] y Lloyd [16].

En la literatura se encuentran diversos algoritmos de agrupamiento, como los propuestos por Nanda [17] y Xu [18]. Sin embargo, K-means tiene ventajas distintas comparadas con esos algoritmos. Estas ventajas consisten en: es fácil de implementar y puede ser usado en una amplia variedad de aplicaciones. En consecuencia, está clasificado como el segundo entre los 10 algoritmos de minería de datos más utilizados [3]. Esta característica lo convierte en el método de referencia para las heurísticas reportadas en varias literaturas [2].

De acuerdo con la literatura consultada, cuando se menciona el algoritmo K-means una referencia importante es MacQueen y en algunos casos es Lloyd. En la Tabla 2.1, se presenta una interpretación de estos algoritmos.

Tabla 2.1. Análisis comparativo entre MacQueen y Lloyd

Some Methods for Classification and Analysis of Multivariate Observations MacQueen (1967) [1]	**Least Squares Quantization in PCM Stuart P. Lloyd (1982)** [16]
Dominio del problema: Estadística. El proceso descrito en este artículo es el siguiente: 1) Seleccionar k grupos del conjunto de datos a particionar, los cuales sólo constan de un objeto aleatorio. 2) Agregar cada nuevo objeto al grupo cuya media es más cercana. 3) Una vez que un objeto es agregado a un grupo, se recalcula la media de ese grupo, con el fin de tomar en cuenta ese objeto. 4) El criterio de paro es minimizar una función objetivo.	**Dominio del problema:** Electrónica. El proceso descrito en este artículo es el siguiente: 1) Seleccionar k grupos aleatoriamente del conjunto de datos a particionar. 2) Asignar cada objeto al centroide más cercano. 3) Recalcular los centroides. 4) El criterio de paro para este proceso, es minimizar la función de ruido.

También Morissette [19] hace una comparativa de dichos algoritmos. Bock [20] presenta información relevante sobre el origen del algoritmo K-means.

2.2. Mejoras del algoritmo en la fase de inicialización

La inicialización consiste en definir los objetos que serán particionados, el número de grupos y un centroide para cada grupo. Se han propuesto varios métodos para definir los centroides iniciales, como los propuesto por Celebi [21].

Los artículos presentados en la Tabla 2.2, muestran una estrategia para seleccionar los centroides iniciales del algoritmo K-means. Los autores de estas investigaciones consideran que ubicando los centroides en posiciones idóneas, origina que el algoritmo converja rápidamente.

Tabla 2.2. Mejoras del algoritmo en la fase de inicialización

Artículo	Estrategia utilizada	Parámetro que mejora	Datos de prueba
A new algorithm for initial cluster centers in k-means algorithm, Murat [22]	Encontrar los puntos más distantes en un espacio de características. Para lo cual proponen dos ejes y eligen el punto central o la media. Después con base en la distancia euclidiana, buscan el primer centroide más alejado a la media. Para elegir el segundo centroide buscan el punto más alejado al primero y así sucesivamente hasta completar el número de clases definidas.	k (número de grupos)	**Reales** Iris: 150 Wine: 178 Letters: 20,000 Ruspini: 75 Spambase: 4,601
Initialization for K-means clustering using Voronoi diagram, Reddy [23]	Crean un diagrama de Voronoi a partir de un conjunto de datos y usan los puntos que se encuentran en los radios más altos de los círculos Voronoi para encontrar los centroides iniciales.	k (número de grupos)	**Reales** Iris: 150 S. Heart: 187 Wine: 178 Ecoli: 336 St. Heart: 270 P.I. Diabetes: 768 Soybean: 47 B.Tissue: 106

2.3. Mejoras del algoritmo en la fase de convergencia

La convergencia consiste en proponer condiciones de paro para el algoritmo K-means. Se aplican diversas condiciones de convergencia tales como: parar el algoritmo cuando alcanza un cierto número de iteraciones; cuando ya no hay cambios de objetos entre grupos; o cuando la diferencia de los centroides en dos iteraciones consecutivas es más pequeña que un determinado umbral. En la Tabla 2.3, se muestra un artículo que aporta una mejora a esta fase.

Tabla 2.3. Mejoras del algoritmo en la fase de convergencia

Artículo	Estrategia utilizada	Parámetro que mejora	Datos de prueba
Improving the Efficiency and Efficacy of the K-means Clustering Algorithm Through a New Convergence Condition, Pérez [24]	La estrategia utilizada consiste en asociar los valores del error cuadrático a una nueva condición de convergencia, la cual ocurre cuando en dos iteraciones consecutivas el error cuadrático de la última iteración excede al de la iteración precedente.	t (número de iteraciones)	**Reales** 846, 214, 768, 270, 178 y 345

2.4. Mejoras del algoritmo en la fase de clasificación

La clasificación, es la etapa más compleja del algoritmo, consiste en calcular la distancia de cada objeto a todos los centroides para asignarlo al centroide más cercano. En la Tabla 2.4, se muestran algunos de los artículos que aportan mejoras para esta fase.

Tabla 2.4. Mejoras del algoritmo en la fase de clasificación

Artículo	Estrategia utilizada	Parámetro que mejora	Datos de prueba
Improvement of the k-means clustering filtering algorithm, Lai [6]	Primero clasifican los grupos en estáticos y activos. Posteriormente usan información de los desplazamientos de los centroides para determinar el conjunto de candidatos para cada nodo de un árbol binario.	k (número de grupos)	**Reales** Block de imágenes de 4x4 pixeles. Utilizan 6 imágenes: Peppers, Lena, Baboon, Parrot, Airplane e Island. **Sintéticos** 20,000
An accelerated K-means clustering algorithm using selection and erasure rules, Lee [25]	Los autores proponen una modificación al algoritmo propuesto por Fahim [15], planteando dos reglas: de selección, usada para adquirir buenos candidatos como centroides iniciales; otra de eliminación, usada para descartar uno o más centroides no calificados hasta que es satisfecha una condición. Para lo anterior se aplica un razonamiento matemático.	k (número de grupos)	**Reales** 20,000 98,304 6,435 12,800
An efficient enhanced k-means clustering algorithm, Fahim [15]	Los autores plantean la siguiente pregunta: ¿Por qué no beneficiarse de la iteración anterior del algoritmo K-means? Se puede mantener la distancia a los centroides más cercanos para cada objeto y en la siguiente iteración calcular la distancia a los grupos anteriores más cercanos. Si la nueva distancia es menor o igual a la distancia anterior, los objetos permanecen en esos grupos y no hay necesidad de calcular sus distancias a los otros grupos.	n (número de objetos)	**Reales** Letters:20,000 Abalone: 4,177 Wind:6,574 **Sintéticos** 10,000 20,000 30,000 40,000 50,000 60,000 70,000 80,000 90,000

Early Classification: A New Heuristic to Improve the Classification Step of K-means, Pérez [7]	Utilizan dos conceptos llamados índice de equidistancia y umbral de equidistancia con el objetivo principal de identificar aquellos objetos con poca probabilidad de cambio de grupo. Para ello plantean las siguientes condiciones: a) Un objeto tiene una alta probabilidad de cambio de grupo si su índice es menor o igual al umbral. b) Un objeto tiene una baja probabilidad de cambio de grupo si su índice es mayor al umbral.	n (número de objetos) t (número de iteraciones)	**Reales** Iris: 150 Concrete compressive strenght: 1,030 Skin segmentation: 245,057 **Sintéticos** 2,500 10,000 40,000
A time-efficient pattern reduction algorithm for k-means clustering, Tsai [5]	Plantean dos métodos: El primero lo utilizan para comprimir y remover objetos; el segundo para asignar los objetos a los centroides más cercanos y para actualizarlos. La idea consiste en revisar el mejor momento para iniciar la compresión y eliminación de objetos hasta que no exceda un límite establecido y así evitar cálculos redundantes.	n (número de objetos)	**Reales** 8,284 95,413 10,000 150 600 6,000 60,000 600,000 6,000,000 10,000,000 **Sintéticas** 400 579 800
A fast k-means clustering algorithm using cluster center displacement, Lai [14]	Se basan en los desplazamientos de los centroides para identificar grupos estáticos y activos. Una vez realizado se calcula la distancia sólo con los centroides activos más cercanos a los objetos.	n (número de objetos)	**Reales** Bloques de imágenes de 4x4 pixeles. **Sintéticas** 10,000 20,000

Improvement to the K-Means Algorithm Through a Heuristics Based on a Bee Honeycomb Structure, Pérez [13]	Plantean que entre una iteración y otra un objeto sólo puede cambiar de membresía a un grupo vecino adyacente. Para ello, realizan ocho cálculos de distancia a sus centroides vecinos para cada objeto.	k (número de grupos)	**Reales** Skin Segmentation D31 **Sintéticas** 2,500 10,000 40,000
Propuesta de tesis (N-means)	Derivado de la observación de las ejecuciones del algoritmo K-means se encontró que algunos grupos se estabilizan primero que otros. Esto es, ya no intercambian objetos con otros grupos. Con base en este conocimiento se desarrolla una nueva heurística a la que se le denominó *grupos estables*, en la cual se descartan los objetos asignados a un grupo estable en las iteraciones posteriores. En consecuencia, se desarrolla una meta-heurística denominada *N-means*, la cual integra las heurísticas *grupos estables* y *Early Classification.*	n (número de objetos) t (número de iteraciones)	**Reales** 245,057 414,528 657,308 **Sintéticas** 10,000 20,000 40,000

Capítulo 3
Meta-heurística N-means

"*Frecuentemente, cuando se termina una pieza de investigación, se ve que han surgido nuevos problemas, nuevos temas y nuevas cuestiones como resultado de los que originalmente se habían considerado en el trabajo de investigación*"

Pauline Young (1960)

En este capítulo se expone la meta-heurística N-means desarrollada en esta investigación. Posteriormente se presentan las heurísticas que la integran y se describe cada una de ellas. El contenido del capítulo se organiza de la siguiente manera: en la Sección 3.1 se muestra el plan de trabajo. En la Sección 3.2 se explica N-means. En la Sección 3.2.1 se explica la heurística *Early Classification*. En la Sección 3.2.2 se expone la heurística grupos estables. Finalmente, en la Sección 3.3 se muestra el algoritmo N-means.

3.1. Plan de trabajo

De manera general, a continuación se muestran las principales actividades realizadas para el desarrollo de la meta-heurística N-means. Dichas actividades se presentan de manera cronológica.

1) Revisión del estado del arte.
2) Estudio y análisis de la heurística *Early Classification*.
3) Estudio y análisis visual del comportamiento del algoritmo K-means. Como resultado de este análisis se obtuvo la heurística denominada *grupos estables.*
4) Propuesta de mejora con base en los resultados obtenidos en los pasos 2 y 3. La unión de las heurísticas *Early Classification* y *grupos estables* integró la nueva meta-heurística N-means.
5) Validación experimental de la nueva meta-heurística N-means con instancias sintéticas y reales.
6) Análisis de los resultados para instancias sintéticas y reales.
7) Conclusiones de la investigación.

3.2. Meta-heurística N-means

Para desarrollar el algoritmo que da origen a la meta-heurística N-means se tomaron como base las heurísticas *EC* y *grupos estables*, tal como se representa en la Figura 3.1. En los siguientes apartados se describirá en que consiste cada una de ellas, así mismo su funcionamiento.

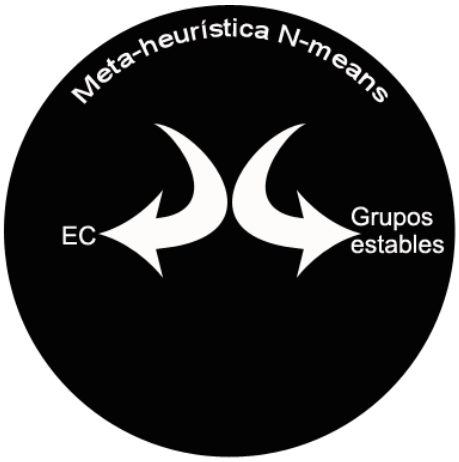

Figura 3.1. Representación gráfica de la meta-heurística N-means.

3.2.1. Heurística Early Classification (EC)

El objetivo principal de la heurística EC es reducir el número de cálculos de distancia en la etapa de clasificación del algoritmo K-means. Esta reducción se logra al aplicar dos conceptos llamados: *índice de equidistancia* y *umbral de equidistancia*. Con base en estos dos conceptos se identifica y excluye de futuros cálculos de distancia aquellos objetos que están localizados fuera de un umbral establecido [7]. En los siguientes párrafos se explicarán los conceptos de *índice de equidistancia* y *umbral de equidistancia*.

Índice de equidistancia

Se define como el valor absoluto de la diferencia de la distancia de un objeto i a sus dos centroides más cercanos μ_1 y μ_2 [7], tal como lo indica la ecuación (1).

$$\alpha_i = abs(\|i - \mu_1\|^2 - \|i - \mu_2\|^2) \quad (1)$$

En la Figura 3.2, se muestra un ejemplo para representar el índice de equidistancia de un objeto i en un plano de dos dimensiones. Por ejemplo, supónganse que dicho objeto tiene las siguientes coordenadas $i\,(3.4, 2.5)$ y sus

dos centroides más cercanos son $\mu_1(2,2)$ y $\mu_2(5,2)$. Por lo tanto, el índice de equidistancia para el objeto i (α_i) se calcula como se muestra a continuación.

$$\alpha_i = abs\big(((3.4-2)^2+(2.5-2)^2)-((3.4-5)^2+(2.5-2)^2)\big) = 0.6$$

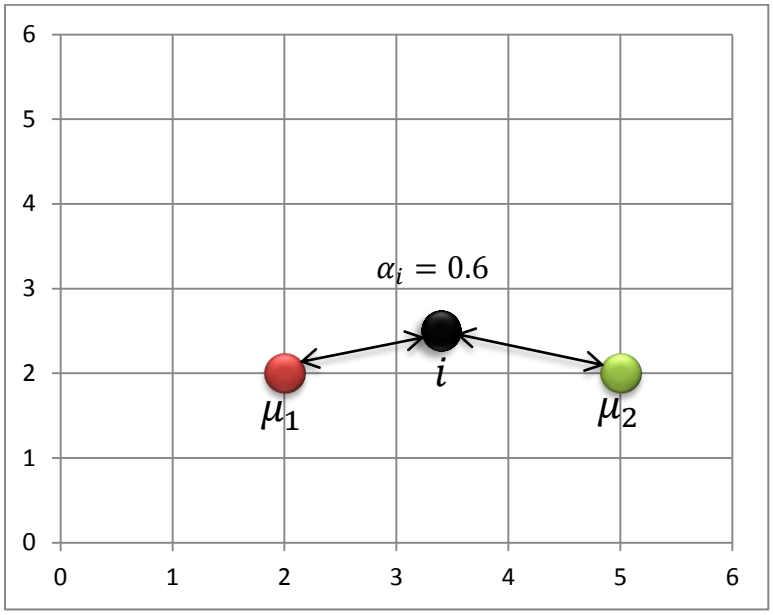

Figura 3.2. Índice de equidistancia.

Umbral de equidistancia

Este concepto permite identificar aquellos objetos con una alta probabilidad de cambio de grupo. El umbral de equidistancia se define como la suma de los dos desplazamientos más grandes (ver ecuación 2) de los centroides μ_x y μ_y en la iteración j ($j > 2$) [7].

$$\beta_j = m_1 + m_2 \qquad (2)$$

Por ejemplo, si en la iteración 3 y 4 se obtuvieran los centroides mostrados en la Tabla 3.1, entonces el valor de los desplazamientos es el que se muestra en la Tabla 3.2, con base en las siguientes expresiones $m_1 = \left\|\mu_{x,j-1} - \mu_{x,j}\right\|^2$ y $m_2 = \left\|\mu_{y,j-1} - \mu_{y,j}\right\|^2$.

Tabla 3.1. Centroides calculados en las iteraciones 3 y 4

Centroides calculados en la iteración 3		Centroides calculados en la iteración 4	
μ_1	$(3.6,5.2)$	μ_1	$(3.5,5.2)$
μ_2	$(4.5,2)$	μ_2	$(4.5,2)$
μ_3	$(6,2.5)$	μ_3	$(5.8,2.8)$
μ_4	$(1.90,2.36)$	μ_4	$(1.90,2.36)$

Tabla 3.2. Desplazamientos calculados

Desplazamientos	
m_1	0.01
m_2	0
m_3	0.13
m_4	0

Por lo tanto, el umbral para la iteración cuatro se calcula como se muestra a continuación.

$$\beta_4 = 0.01 + 0.13 = 0.14$$

Una vez calculados el índice de equidistancia y el umbral de equidistancia, el siguiente paso es localizar aquellos objetos que cumplen las siguientes condiciones:

a) Cuando el índice de equidistancia es mayor al umbral de equidistancia $(\alpha_i > \beta_j)$, (ver Figura 3.3, que se localiza en la pág. 18), se considera que el objeto *i* tiene una baja probabilidad de cambio de grupo, por lo cual se descarta de futuros cálculos de distancias en iteraciones posteriores.

b) Cuando el índice de equidistancia es menor o igual al umbral de equidistancia $(\alpha_i \leq \beta_j)$, (ver Figura 3.4, que se localiza en pág. 18), se considera que el objeto *i* tiene una alta probabilidad de cambio de grupo, por lo cual es considerado para futuros cálculos de distancia en iteraciones posteriores.

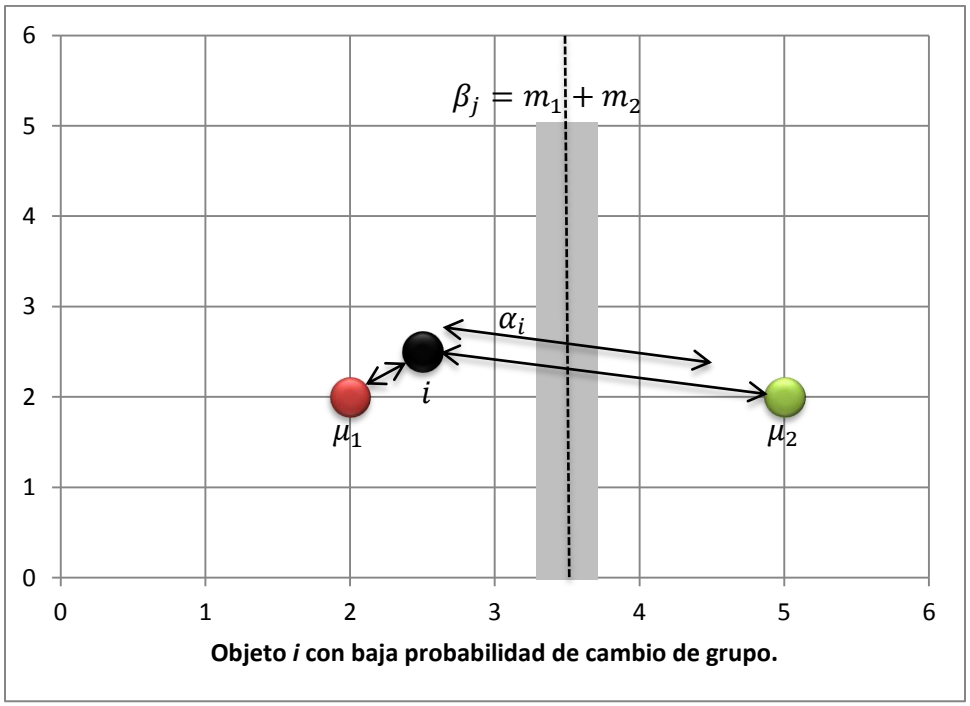

Figura 3.3. Representación gráfica cuando el índice de equidistancia es mayor al umbral.

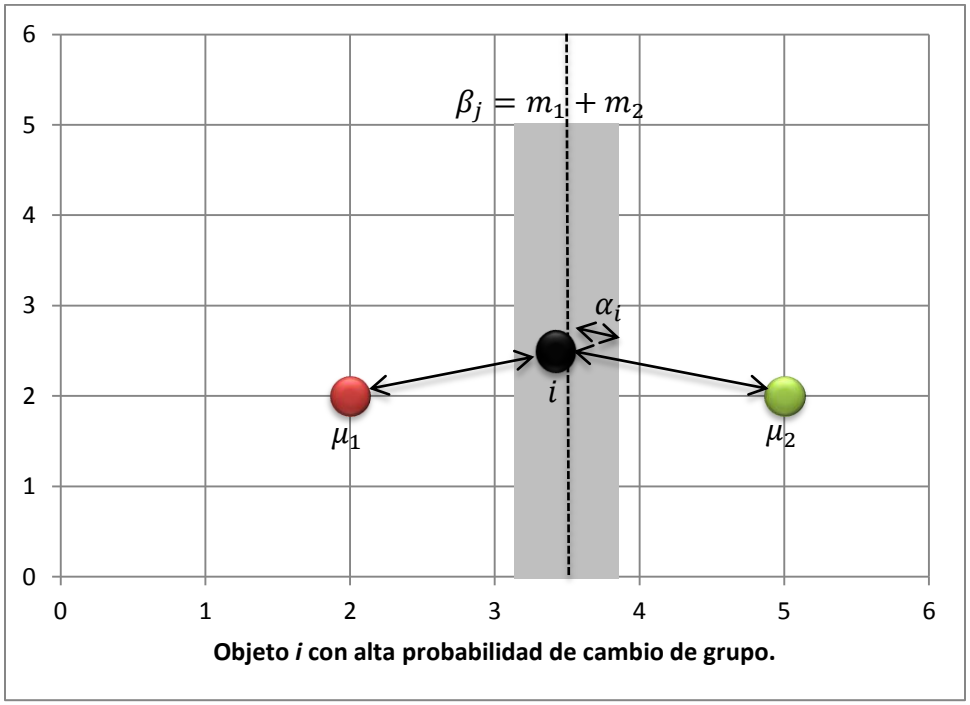

Figura 3.4. Representación gráfica cuando el índice de equidistancia es menor o igual al umbral.

Para conocer mayores detalles de la heurística EC se invita al lector a consultar Pérez [7].

3.2.2. Heurística grupos estables

La heurística *grupos estables* se deriva de la observación de las ejecuciones del algoritmo K-means. A partir de dicha observación se identifica que algunos grupos se estabilizan primero que otros. De manera que un grupo estable es aquel que ya no tiene intercambio de objetos con otros grupos en iteraciones posteriores.

Para exponer dicha heurística se utilizó una instancia de 10,000 objetos divididos en 100 grupos con una distribución uniforme de dos dimensiones en un espacio de 100 x 100. Los centroides iniciales se generaron aleatoriamente y el criterio de convergencia fue que ya no ocurriera intercambio de objetos entre grupos.

El número total de iteraciones que realiza el algoritmo son 16, de las cuales a partir de la iteración ocho, se identifican los primeros grupos que cumplen con la condición de grupos estables. A continuación en las Figuras 3.5 a 3.13 (que se localizan en las págs. 21 a 29), se muestran las últimas nueve iteraciones del algoritmo cuando resuelve la instancia de 10,000 objetos.

En la Figura 3.5, se muestra el resultado de agrupamiento de la iteración ocho. Como se observa, los cien grupos se identifican con un punto de color azul encerrados en líneas de color negro. Asimismo, se observa que a partir de esta iteración, se identifica el primer grupo que cumple con la condición de grupos estables, para este caso, el encerrado con líneas rojas y con el identificador de un número uno. Por lo que se concluye que los objetos que pertenecen a dicho grupo se excluyen de futuros cálculos de distancia.

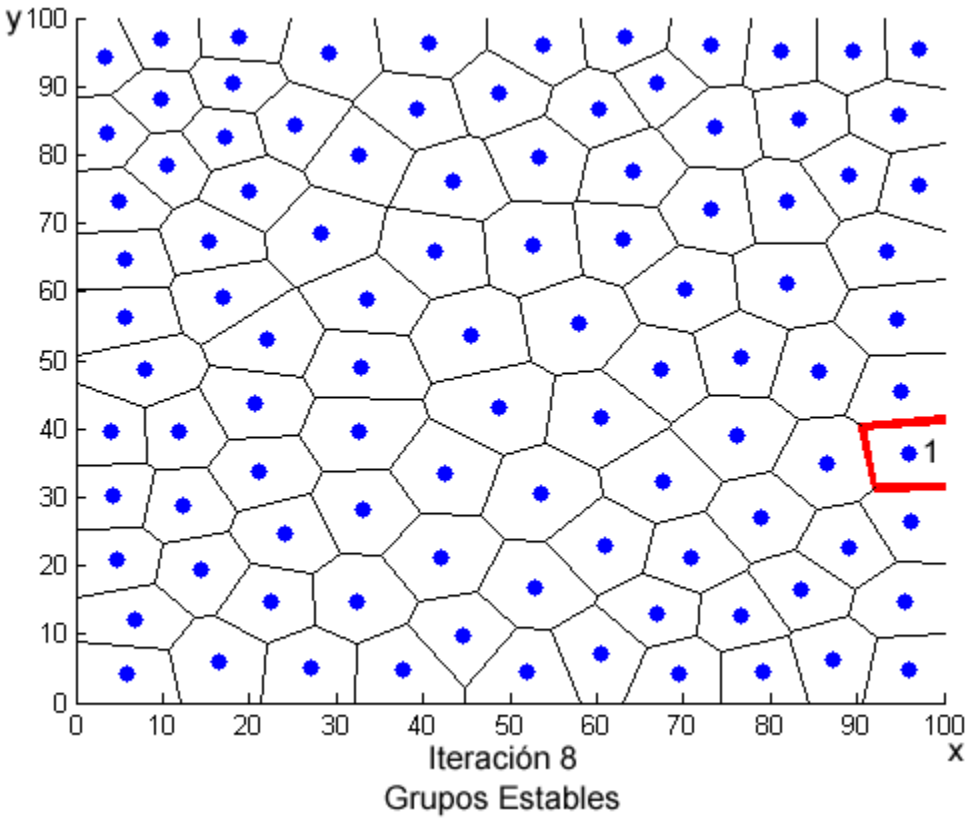

Figura 3.5. Grupos estables identificados en la iteración 8.

En la Figura 3.6 (que se localiza en la pág. 22), se muestra el resultado de agrupamiento de la iteración nueve. Como se observa los 2 grupos que cumplen con la condición de grupos estables son: los encerrados con líneas de color verde y con los identificadores 2 y 3. Debido a que en la iteración previa se encontró el primer grupo estable al cual se colocó el identificador 1 se optó por colocar los identificadores 2 y 3.

De este conocimiento se concluye que los objetos que pertenecen a los grupos con identificadores 2 y 3 se excluyen de futuros cálculos de distancia para iteraciones posteriores.

Para comprender mejor esta iteración (Figura 3.6) se excluye el grupo identificado previamente, permaneciendo sólo los grupos que cumplen con la condición de grupos estables para la presente iteración.

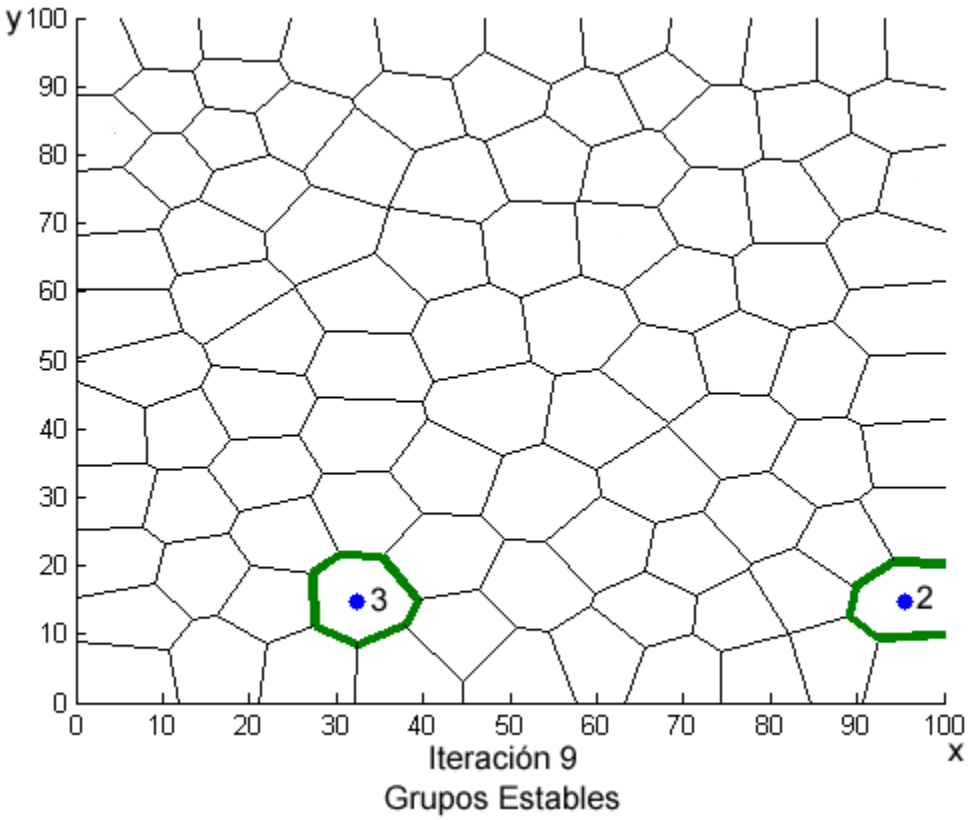

Figura 3.6. Grupos estables identificados en la iteración 9.

En la Figura 3.7 (que se localiza en la pág. 23), se muestra el resultado de agrupamiento de la iteración 10. Como se observa los 8 grupos que cumplen con la condición de grupos estables son: los encerrados con líneas de color rosa y con los identificadores del 4 al 11. Debido a que en la iteración previa se encontraron 2 grupos estables a los cuales se les colocó los identificadores 2 y 3 se optó por colocar los identificadores del 4 al 11.

De este conocimiento se concluye que los objetos que pertenecen a los grupos con identificadores del 4 al 11 se excluyen de futuros cálculos de distancia para iteraciones posteriores.

Para comprender mejor esta iteración (Figura 3.7) se omiten los grupos que se identificaron previamente, permaneciendo sólo los grupos que cumplen con la condición de grupos estables para la presente iteración.

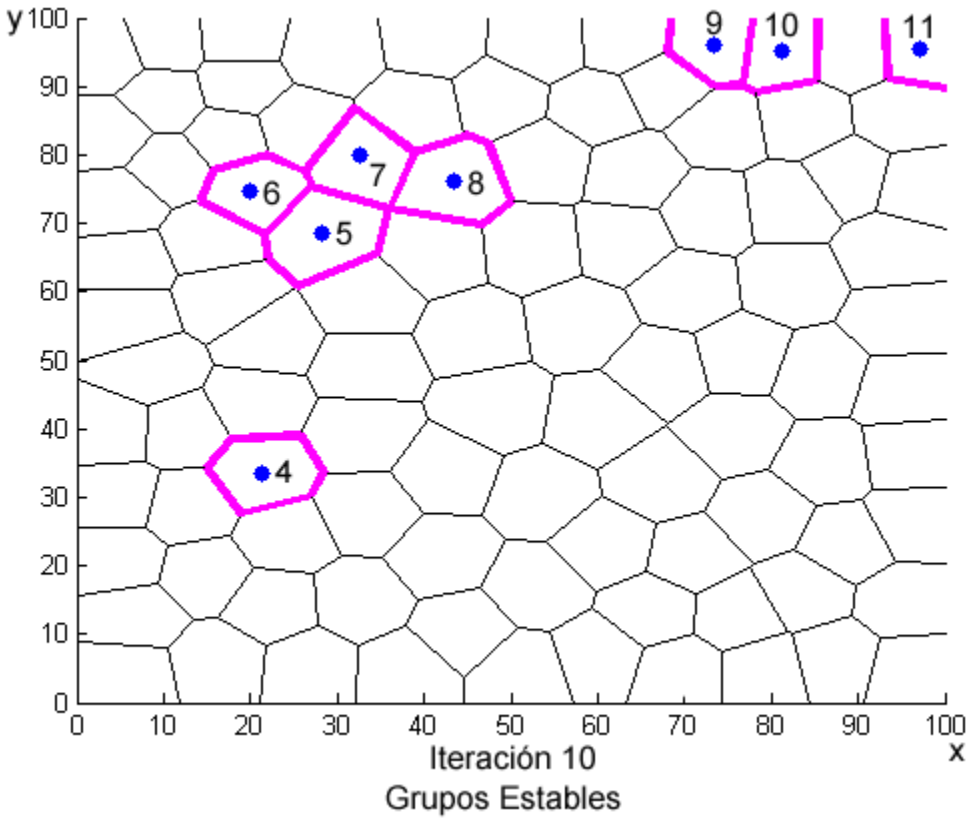

Figura 3.7. Grupos estables identificados en la iteración 10.

En la Figura 3.8 (que se localiza en la pág. 24), se muestra el resultado de agrupamiento de la iteración 11. Como se observa los 20 grupos que cumplen con la condición de grupos estables son: los encerrados con líneas de color amarillo y con los identificadores del 12 al 31. Debido a que en la iteración previa se encontraron 8 grupos estables a los cuales se les colocó los identificadores del 4 al 11 se optó por colocar los identificadores del 12 al 31.

De este conocimiento se concluye que los objetos que pertenecen a los grupos con identificadores del 12 al 31 se excluyen de futuros cálculos de distancia para iteraciones posteriores.

Para comprender mejor esta iteración (Figura 3.8) se omiten los grupos identificados previamente, permaneciendo sólo los grupos que cumplen con la condición de grupos estables para la presente iteración.

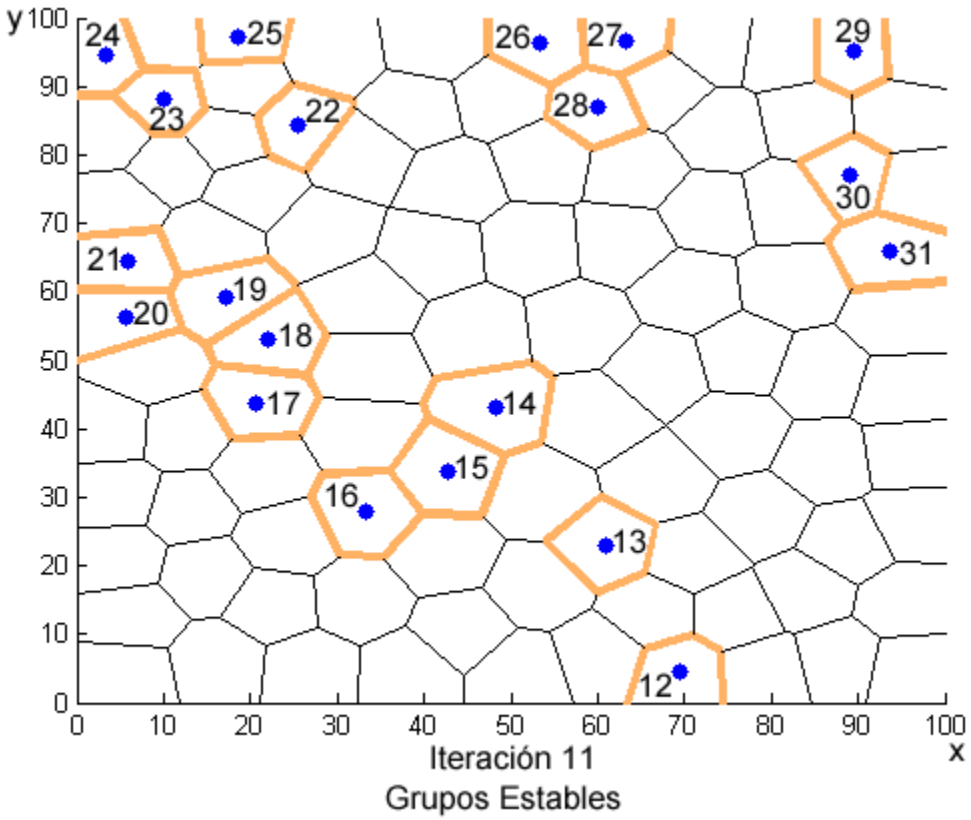

Figura 3.8. Grupos estables identificados en la iteración 11.

En la Figura 3.9 (que se localiza en la pág. 25), se muestra el resultado de agrupamiento de la iteración 12. Como se observa los 21 grupos que cumplen con la condición de grupos estables son: los encerrados con líneas de color morado y con los identificadores del 32 al 52. Debido a que en la iteración previa se encontraron 20 grupos estables a los cuales se les colocó los identificadores del 12 al 31 se optó por colocar los identificadores del 32 al 52.

De este conocimiento se concluye que los objetos que pertenecen a los grupos con identificadores del 32 al 52 se excluyen de futuros cálculos de distancia para iteraciones posteriores.

Para comprender mejor esta iteración (Figura 3.9) se omiten los grupos identificados previamente, permaneciendo sólo los grupos que cumplen con la condición de grupos estables para la presente iteración.

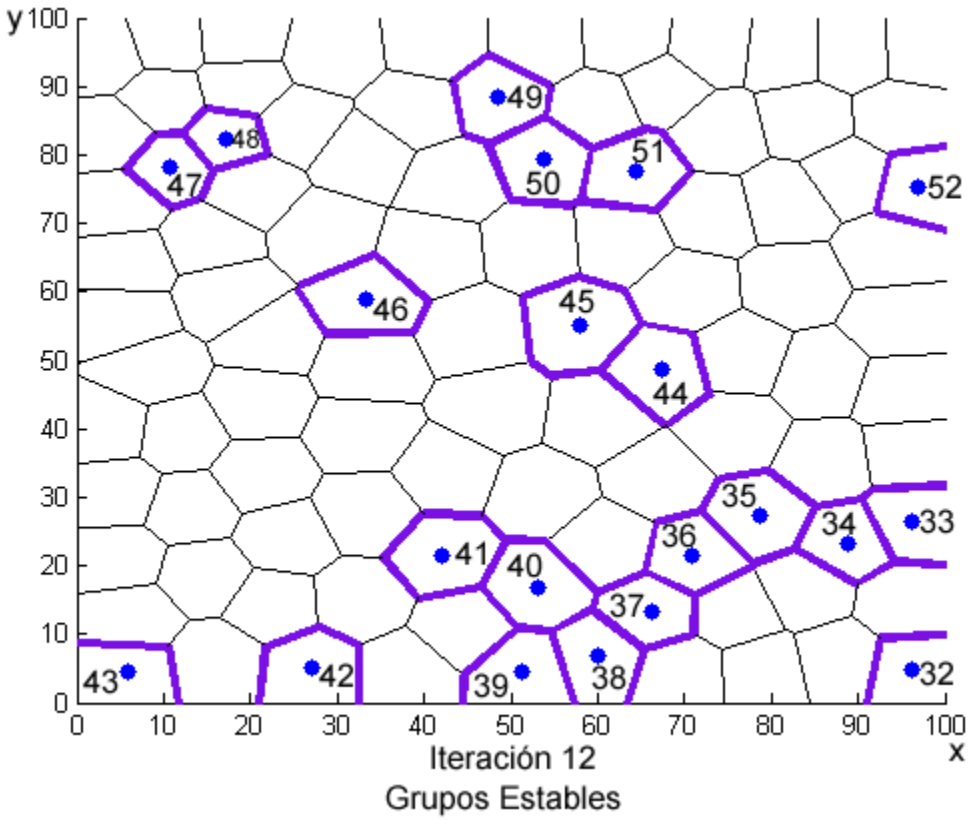

Figura 3.9. Grupos estables identificados en la iteración 12.

En la Figura 3.10 (que se localiza en la pág. 26), se muestra el resultado de agrupamiento de la iteración 13, como se observa los 23 grupos que cumplen con la condición de grupos estables son: los encerrados con líneas de color azul y con los identificadores del 53 al 75. Debido a que en la iteración previa se encontraron 21 grupos estables a los cuales se les colocó los identificadores del 32 al 52 se optó por colocar los identificadores del 53 al 75.

De este conocimiento se concluye que los objetos que pertenecen a los grupos con identificadores del 53 al 75 se excluyen de futuros cálculos de distancia para iteraciones posteriores.

Para comprender mejor esta iteración (Figura 3.10) se omiten los grupos previamente identificados, permaneciendo sólo los grupos que cumplen con la condición de grupos estables para la presente iteración.

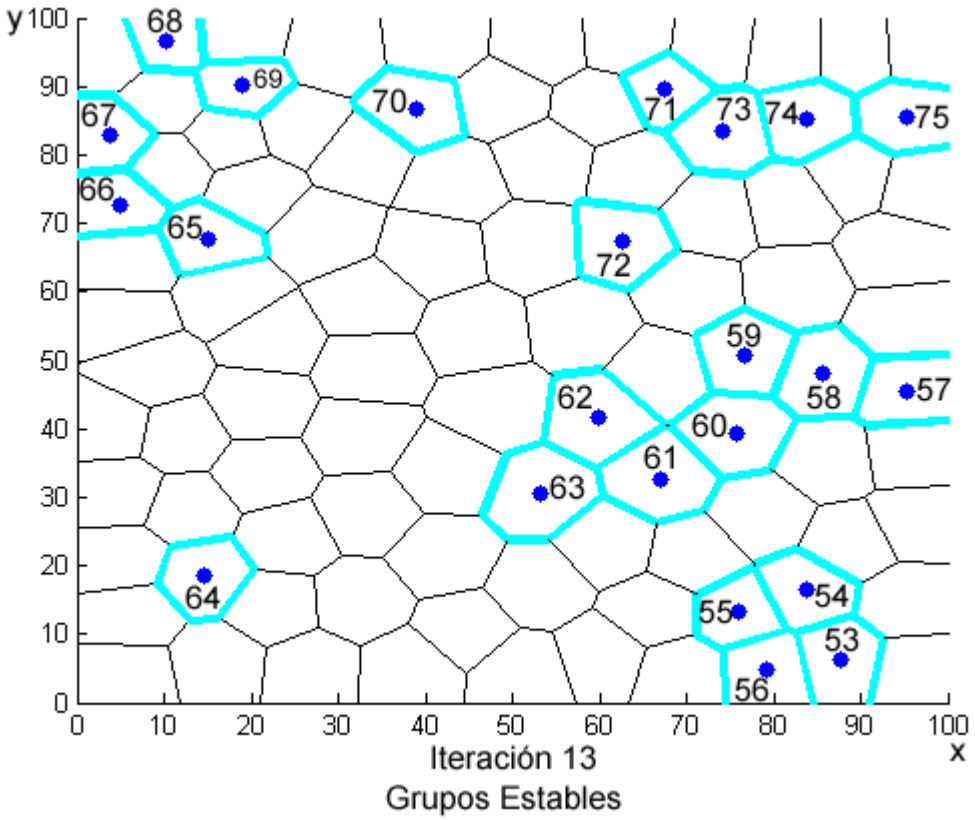

Figura 3.10. Grupos estables identificados en la iteración 13.

En la Figura 3.11 (que se localiza en la pág. 27), se muestra el resultado de agrupamiento de la iteración 14, como se observa los 13 grupos que cumplen con la condición de grupos estables son: los encerrados con líneas de color café y con los identificadores del 76 al 88. Debido a que en la iteración previa se encontraron 23 grupos estables a los cuales se les colocó los identificadores del 53 al 75 se optó por colocar los identificadores del 76 al 88.

De este conocimiento se concluye que los objetos que pertenecen a los grupos con identificadores del 76 al 88 se excluyen de futuros cálculos de distancias para iteraciones posteriores.

Para comprender mejor esta iteración (Figura 3.11) se omiten los grupos previamente identificados, permaneciendo sólo los grupos que cumplen con la condición de grupos estables para la presente iteración.

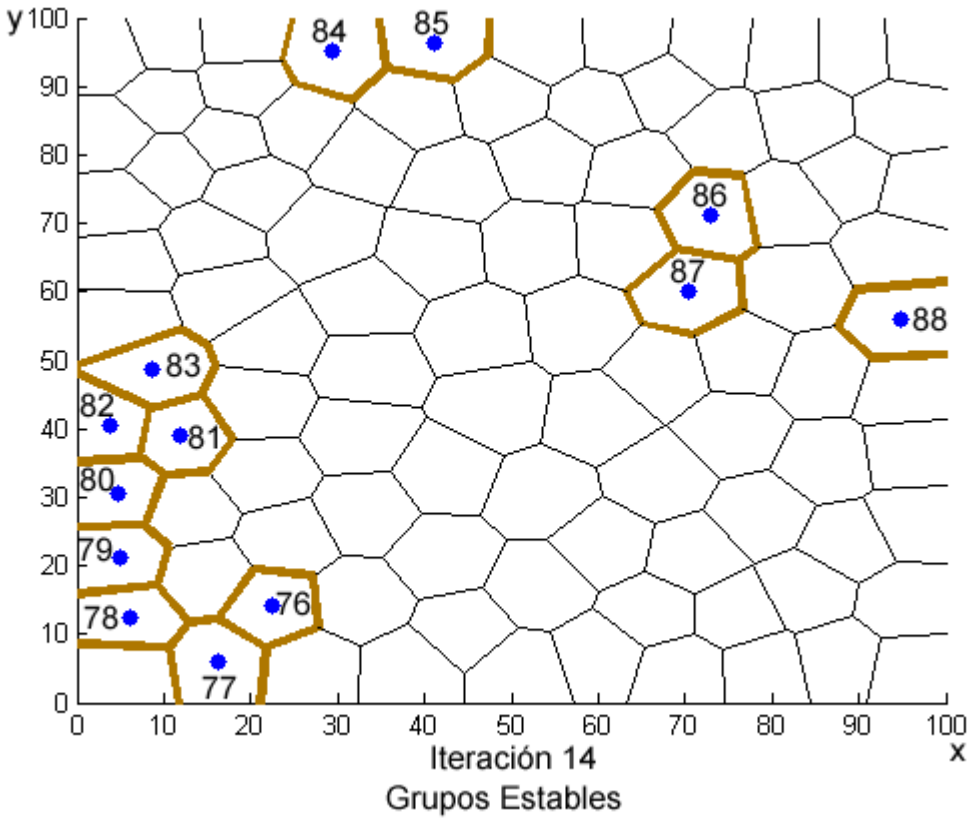

Figura 3.11. Grupos estables identificados en la iteración 14.

En la Figura 3.12 (que se localiza en la pág. 28), se muestra el resultado de agrupamiento de la iteración 15, como se observa los 9 grupos que cumplen con la condición de grupos estables son: los encerrados con líneas de color azul y con los identificadores del 89 al 97. Debido a que en la iteración previa se encontraron 13 grupos estables a los cuales se les colocó los identificadores del 76 al 88 se optó por colocar los identificadores del 89 al 97.

De este conocimiento se concluye que los objetos que pertenecen a los grupos con identificadores del 89 al 97 se excluyen de futuros cálculos de distancia para iteraciones posteriores.

Para comprender mejor esta iteración (Figura 3.12) se omiten los grupos previamente identificados, permaneciendo sólo los grupos que cumplen con la condición de grupos estables para la presente iteración.

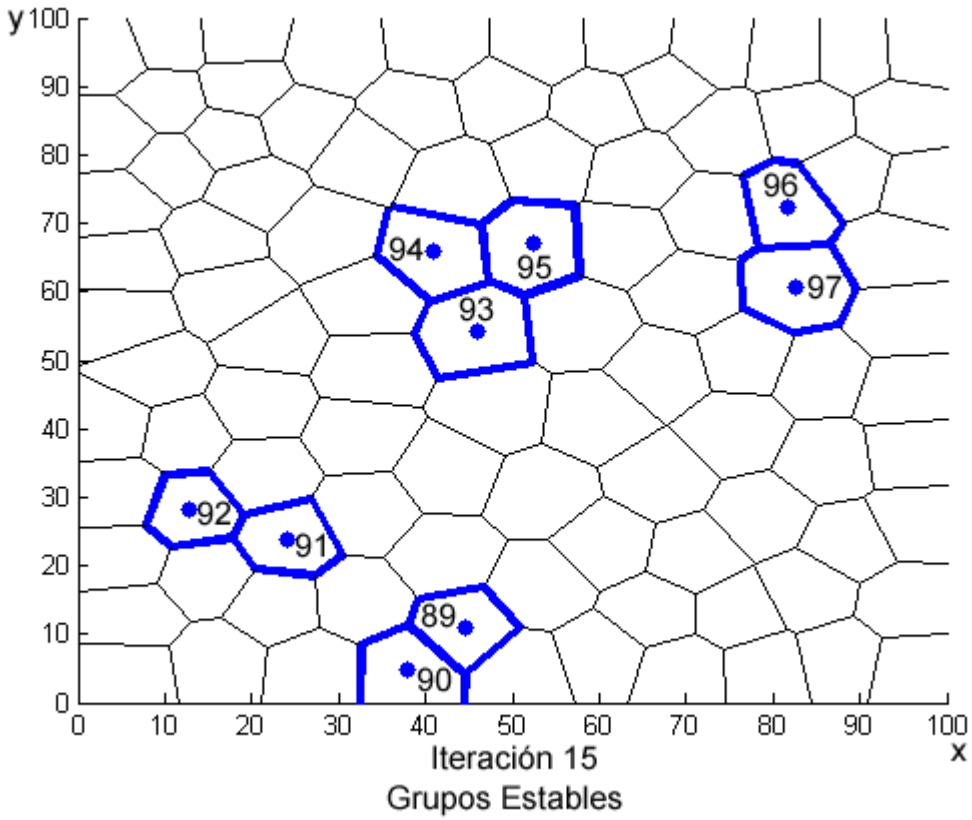

Figura 3.12. Grupos estables identificados en la iteración 15.

En la Figura 3.13 (que se localiza en la pág. 29), se muestra el resultado de agrupamiento de la última iteración (iteración 16). Como se observa los 3 grupos que cumplen con la condición de grupos estables son: los encerrados con líneas de color verde y con los identificadores del 98 al 100. Debido a que en la iteración previa se encontraron 9 grupos estables a los cuales se les colocó los identificadores del 89 al 97 se optó por utilizar los identificadores del 98 al 100. Como se observa (Figura 3.13) los 100 grupos cumplen con la condición de grupos estables, en consecuencia el algoritmo converge.

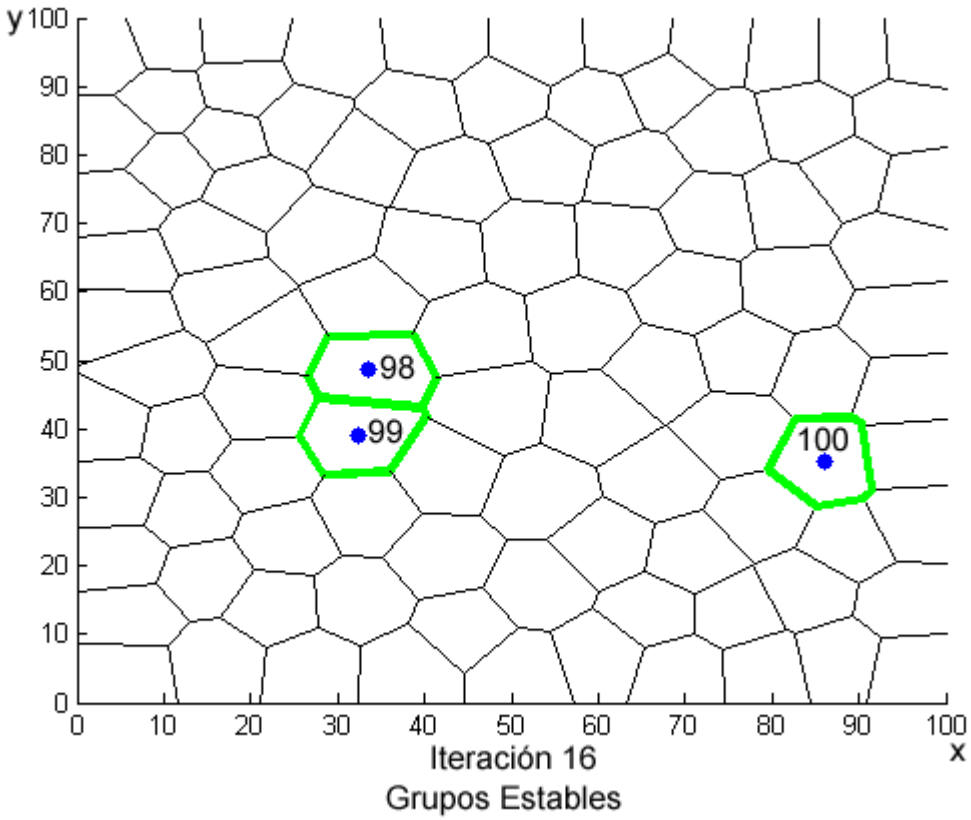

Figura 3.13. Grupos estables identificados en la iteración 16.

Con base en este conocimiento se desarrolló la heurística *grupos estables*, la cual establece que los objetos asignados a un grupo estable se descartan de los cálculos de distancia en las iteraciones posteriores.

3.3. Algoritmo N-means

En la Tabla 3.3, se muestra el algoritmo de la meta-heurística N-means, el cual tiene como entradas un conjunto de datos D, el número de objetos n, el número de grupos k y las dimensiones d. Como salidas tiene los grupos finales C y el valor de los centroides finales M^t.

Tabla 3.3. Algoritmo de la meta-heurística N-means

	Algoritmo N-means
	Entradas $\boldsymbol{D}$, $\boldsymbol{n}$, $\boldsymbol{k}$, $\boldsymbol{d}$ **Salidas** $\boldsymbol{C}, \boldsymbol{M^t}$
1	Elegir k objetos del conjunto D aleatoriamente y asignarlos al arreglo de centroides iniciales $M^0 = \{{m_1}^0, \dots, {m_k}^0\}$
2	Establecer $t \leftarrow 0$; $determinante[n] \leftarrow 0$;
3	**Hacer**
4	**Para** $i \leftarrow 1$ **Hasta** n **Hacer**
5	**Si** $(determinante[i] = 1 \ \vee \ t \leq 2)$ **Entonces**
6	**Para** $j \leftarrow 1$ **Hasta** k **Hacer**
7	$distancia \leftarrow d(x_i, m_j)$;
8	$cluster[i] \leftarrow$ Almacenar el identificador j del centroide más cercano para el objeto x_i;
9	**Fin para**
10	**Fin si**
11	**Fin para**
12	Hacer el cálculo de los nuevos centroides M^t
13	**Para** $j \leftarrow 1$ **Hasta** k **Hacer**
14	**Si** $(t \geq 2)$ **Entonces**
15	$desplazamientos[j] \leftarrow d\left(M^{(t-1)}, M^t\right)$;
16	**Fin si**
17	**Fin para**
18	**Si** $(t \geq 2 \ \wedge \ M^{(t-1)} \neq M^t)$ **Entonces**
19	**Para** $j \leftarrow 1$ **Hasta** k **Hacer**
20	$umbral \leftarrow desplaza_{j1} + desplaza_{j2}$;
21	**Fin para**
22	**Para** $i \leftarrow 1$ **Hasta** n **Hacer**
23	**Si** $(determinante[i] = 1 \ \vee \ t = 2)$ **Entonces**
24	**Para** $j \leftarrow 1$ **Hasta** k **Hacer**
25	$cluster2[i] \leftarrow j$;
26	**Fin para**
27	$indice \leftarrow abs(\|x_i - cluster[i]\|^2 - \|x_i - cluster2[i]\|^2)$;
28	**Si** $\left(indice > umbral \ \vee \ desplazamiento\left[cluster[i]\right] = 0\right)$ **Entonces**
29	$determinante[i] \leftarrow 0$;
30	**Fin si**
31	**Sino Entonces**
32	$determinante[i] \leftarrow 1$;
33	**Fin sino**
34	**Fin si**
35	**Fin para**
36	**Fin si**
37	$t \leftarrow t + 1$;
38	**Mientras** $(M^{(t-1)} \ \neq M^t)$;

En la línea 1 se definen los centroides iniciales, los cuales se eligen aleatoriamente del conjunto de datos a particionar. Esta instrucción sólo se realiza en la primera iteración.

En la línea 2 se inicializan un contador de iteraciones y una estructura de datos de tamaño $n * 1$. En la línea 3 se inicia un ciclo que termina en la línea 38. En esta línea se define el criterio de paro del algoritmo.

En las líneas 4 a 11 se realiza el cálculo de distancia de cada objeto a los centroides. Este cálculo se realiza en las primeras tres iteraciones para inicializar la heurística. Posteriormente se determinan los objetos que tienen una alta y baja probabilidad de cambio de grupo para identificar cuáles se van a calcular.

En la línea 12 se realiza el recálculo de los centroides. En las líneas 13 a 17 se inicia un ciclo para calcular los desplazamientos de los centroides. Dicho cálculo se realiza a partir de la tercera iteración.

En las líneas 18 a 36 se realiza el cálculo del umbral y el índice de equidistancia. En este caso, el cálculo se realiza a partir de la tercera iteración y mientras los centroides sean distintos.

En las líneas 19 a 21 comienza un ciclo para calcular el umbral de equidistancia que consiste en buscar los dos desplazamientos más grandes.

En las líneas 22 a 35 se inicia un ciclo para calcular el índice de equidistancia. Este índice se obtiene calculando el valor absoluto de la diferencia de las distancias de un objeto a sus dos centroides más cercanos.

En las líneas 28 a 33 se identifican las condiciones que propone la meta-heurística, las cuales son las siguientes:

a) Un objeto tiene una baja probabilidad de cambio de grupo si el índice es mayor al umbral o si dicho objeto pertenece a un grupo que cumple con la condición de grupos estables.
b) Un objeto tiene una alta probabilidad de cambio de grupo si el índice es menor o igual al umbral; o si dicho objeto pertenece a un grupo activo.

Capítulo 4
Validación experimental y análisis de resultados

"*A los que quisieron y no pudieron.*
A los que lo intentaron, no pudieron y no repitieron.
A los que lo quisieron, lo intentaron, no pudieron, lo repitieron y lo superaron,
porque ellos son los verdaderos vencedores"
Jean-Pierre Lévy Mangin (2006)

En este capítulo se desarrolla la explicación del análisis experimental de la meta-heurística N-means. Esta meta-heurística fue validada con instancias sintéticas y reales. El contenido del capítulo se organiza de la siguiente manera: en la Sección 4.1 se explican cómo se obtuvieron los resultados comparativos con instancias sintéticas. En la Sección 4.2 se presentan los resultados comparativos obtenidos con instancias reales. Finalmente, en la Sección 4.3 se muestran resultados más destacables producto de este análisis.

Para validar la meta-heurística N-means se tomó como base la metodología experimental propuesta por McGeoch [26]. Se implementó N-means y se midió su desempeño con base en dos elementos: tiempo de ejecución y calidad de la solución. El criterio de convergencia aplicado fue cuando los centroides no cambiaron de una iteración previa a la actual.

4.1. Validación experimental con instancias sintéticas

Objetivo del experimento

Observar el comportamiento de *N-means* en contraste con los algoritmos *K-means* y *EC*, cuando se incrementa el número de grupos para instancias sintéticas de 10,000, 20,000 y 40,000 objetos.

Planeación del experimento

a) *Formulación de la pregunta:* ¿Qué tanto reduce *N-means* la complejidad de los algoritmos *K-means*, *EC* y que tanto afecta la calidad de agrupamiento incrementando el número de grupos?
b) *Ambiente de prueba: K-means*, *EC* y *N-means* se programaron en lenguaje C. Características del equipo: Procesador Intel Core 2 Duo T6400 2.0GHz, 4GB en RAM y 500GB en HD. Sistema operativo: Linux (Ubuntu) 13.04. Instancias de prueba: sintéticas de 10,000, 20,000 y 40,000 objetos todas en dos dimensiones.
c) *Diseño del experimento:* para responder la pregunta planteada se incrementa el número de grupos con valores de k=100, 200, 400 y 800. Propiedades a medir: tiempo y calidad de agrupamiento. El número de ejecuciones para cada caso es de 30.

Ejecución del experimento

a) Correr las pruebas y recolectar datos

En las Tablas 4.1 y 4.2, se muestran los resultados promedio de 30 ejecuciones de tiempo y calidad de los algoritmos *K-means*, *EC* y *N-means*

para las instancias sintéticas de 10,000, 20,000 y 40,000 objetos con valores de *k*=100, 200, 400 y 800.

Tabla 4.1. Resultados promedio de 30 ejecuciones para instancias sintéticas (tiempo)

Instancia	Grupos	Tiempo(ms) K-means	Tiempo(ms) EC	Tiempo(ms) N-means
10,000	100	3849.979	739.062	668.082
20,000	100	17499.292	2022.607	1983.519
40,000	100	32701.377	3137.857	2920.595
10,000	200	4602.995	1570.891	1287.015
20,000	200	25660.527	4487.160	4005.721
40,000	200	57240.199	6623.690	6061.431
10,000	400	5883.281	3302.487	2348.602
20,000	400	33135.989	9604.435	7848.134
40,000	400	71962.976	14211.106	12163.231
10,000	800	7536.835	6852.746	4077.238
20,000	800	42475.577	21496.191	14756.262
40,000	800	95042.325	30633.120	23219.624

Tabla 4.2. Resultados promedio de 30 ejecuciones para instancias sintéticas (calidad)

Instancia	Grupos	Calidad K-means	Calidad EC	Calidad N-means
10,000	100	38329.806	39455.579	39470.049
20,000	100	108187.032	111355.157	111364.481
40,000	100	305214.128	315273.131	315280.559
10,000	200	27229.909	27712.241	27754.241
20,000	200	76667.136	78546.191	78579.342
40,000	200	216078.493	222053.831	222079.198
10,000	400	19353.204	19490.477	19580.995
20,000	400	54412.625	55200.226	55307.914
40,000	400	153144.216	156342.739	156431.945
10,000	800	13774.106	13783.417	13870.518
20,000	800	38707.311	38924.159	39112.552
40,000	800	108667.188	110091.526	110326.970

b) Análisis de los datos

En las columnas tres y cuatro de la Tabla 4.3, se muestra la diferencia del tiempo de ejecución para los algoritmos *EC* y *N-means* en contraste con el algoritmo *K-means*. En la última columna se muestra la diferencia de tiempo entre los algoritmos *EC* y *N-means*. Como se observa (ver valores sombreados), cuando se incrementa el número de grupos para cada una de las instancias dadas, *N-means* obtiene mejores resultados en comparación con el algoritmo *EC*.

Tabla 4.3. Diferencias de tiempo entre los algoritmos K-means, EC y N-means

Instancia	Grupos	Diferencia (%) K-Means Vs. EC	Diferencia (%) K-means Vs. N-means	Diferencia (%) EC Vs. N-means
10,000	100	80.80	82.64	9.60
20,000	100	88.44	88.66	1.93
40,000	100	90.40	91.06	6.92
10,000	200	65.87	72.03	18.07
20,000	200	82.51	84.38	10.72
40,000	200	88.42	89.41	8.48
10,000	400	43.86	60.08	28.88
20,000	400	71.01	76.31	18.28
40,000	400	80.25	83.09	14.41
10,000	800	9.07	45.90	40.50
20,000	800	49.39	65.25	31.35
40,000	800	67.76	75.56	24.20

En las columnas tres y cuatro de la Tabla 4.4, se muestran las diferencias de calidad de agrupamiento entre los algoritmos *EC* y *N-means* en contraste con el algoritmo *K-means*. En la última columna se muestra la diferencia en la calidad entre los algoritmos *EC* y *N-means*. Como se observa (ver valores sombreados), la disminución en la calidad no fue significativa.

Tabla 4.4. Diferencias de calidad entre los algoritmos K-means, EC y N-means

Instancia	Grupos	Diferencia (%) K-means Vs. EC	Diferencia (%) K-means Vs. N-means	Diferencia (%) EC Vs. N-means
10,000	100	-2.937	-2.974	-0.036
20,000	100	-2.928	-2.936	-0.008
40,000	100	-3.295	-3.298	-0.002
10,000	200	-1.771	-1.925	-0.151
20,000	200	-2.450	-2.494	-0.042
40,000	200	-2.765	-2.777	-0.011
10,000	400	-0.709	-1.177	-0.464
20,000	400	-1.447	-1.645	-0.195
40,000	400	-2.088	-2.146	-0.057
10,000	800	-0.067	-0.699	-0.631
20,000	800	-0.560	-1.046	-0.483
40,000	800	-1.310	-1.527	-0.213

En las Figuras 4.1, 4.2 y 4.3, se muestran gráficamente los resultados promedio de 30 ejecuciones del tiempo de ejecución para las instancias sintéticas de 10,000, 20,000 y 40,000 objetos. Como se observa a medida que el número de grupos se incrementa la complejidad del algoritmo *N-means* (línea negra) es menor en comparación con los algoritmos *EC* (línea azul) y *K-means* (línea roja) para todos los casos.

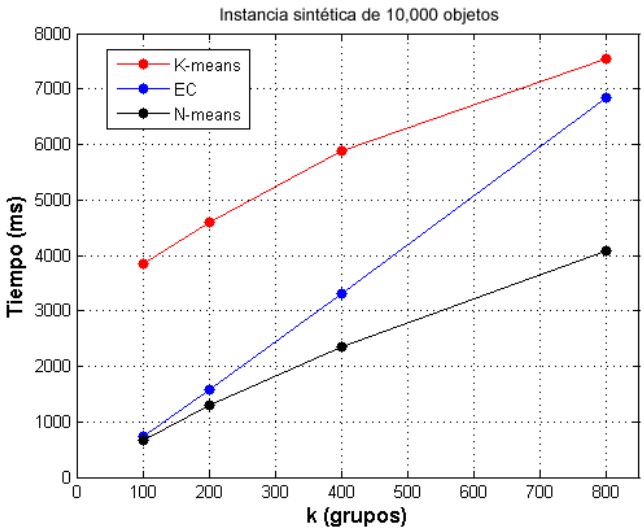

Figura 4.1. Comparación de tiempo de ejecución para una instancia sintética de 10,000 objetos.

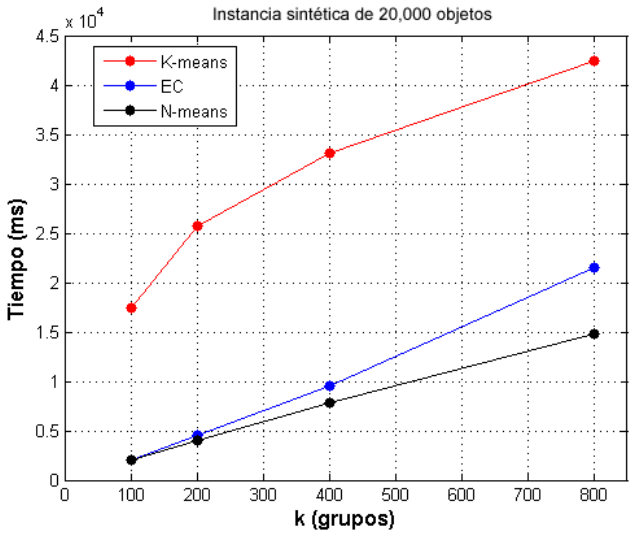

Figura 4.2. Comparación de tiempo de ejecución para una instancia sintética de 20,000 objetos.

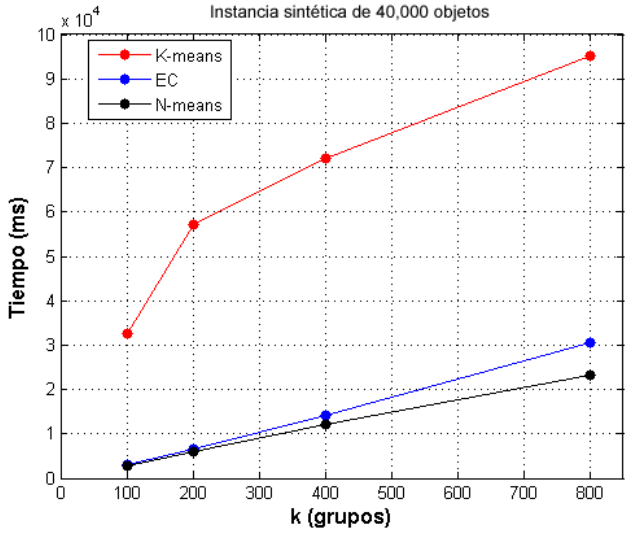

Figura 4.3. Comparación de tiempo de ejecución para una instancia sintética de 40,000 objetos.

En las Figuras 4.4, 4.5 y 4.6, se representan gráficamente los resultados promedio de 30 ejecuciones de la calidad de agrupamiento para las instancias sintéticas de 10,000, 20,000 y 40,000 objetos. Se observa que la disminución de la calidad no fue significativa.

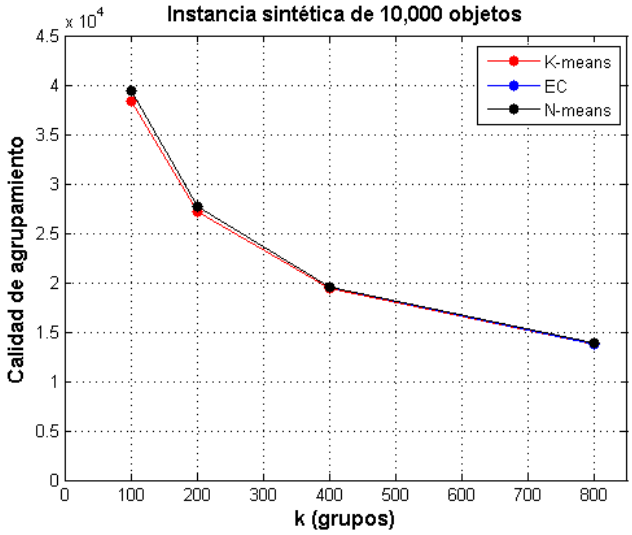

Figura 4.4. Comparación de calidad de agrupamiento para una instancia sintética de 10,000 objetos.

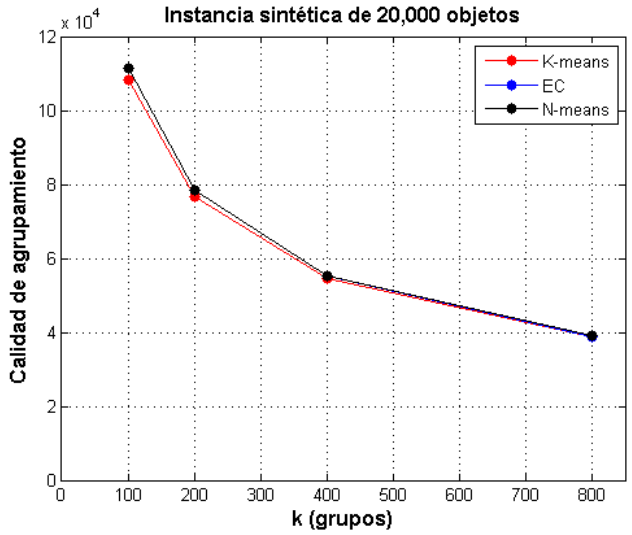

Figura 4.5. Comparación de calidad de agrupamiento para una instancia sintética de 20,000 objetos.

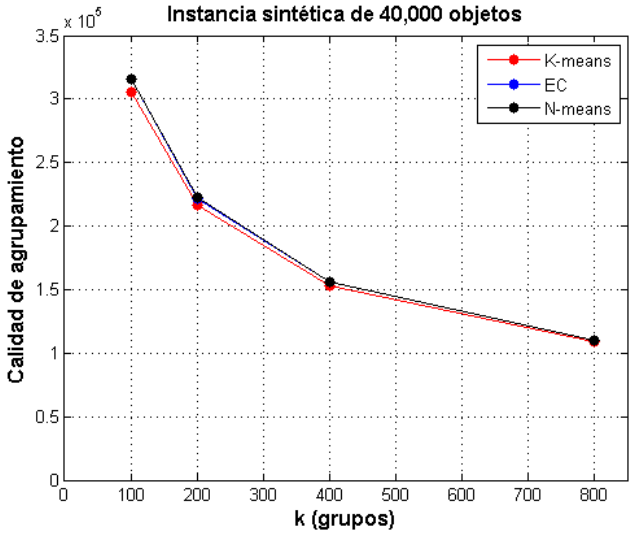

Figura 4.6. Comparación de calidad de agrupamiento para una instancia sintética de 40,000 objetos.

4.2. Validación experimental con instancias reales

Objetivo del experimento

Observar el comportamiento de *N-means* en contraste con los algoritmos *K-means* y *EC*, cuando se incrementa el número de grupos para instancias reales de 245,057, 414,528 y 657,308 objetos.

Planeación del experimento

a) *Formulación de la pregunta:* ¿Qué tanto reduce *N-means* la complejidad de los algoritmos *K-means*, *EC* y que tanto afecta la calidad de agrupamiento incrementando el número de grupos?

b) *Ambiente de prueba: K-means*, *EC* y *N-means* se programaron en lenguaje C. Características del equipo: Procesador Intel Core 2 Duo T6400 2.0GHz, 4GB en RAM y 500GB en HD. Sistema operativo: Linux (Ubuntu) 13.04. Instancias de prueba: instancias reales *Skin Segmentation* de 245,057 objetos en tres dimensiones, *Paris* de 414,528 objetos en dos dimensiones y *New York* de 657,308 objetos en dos dimensiones.

c) *Diseño del experimento:* para responder la pregunta planteada se incrementa el número de grupos con valores de *k*=100, 200, 400 y 800. Propiedades a medir: tiempo y calidad de agrupamiento. El número de ejecuciones para cada caso es de 30.

Ejecución del experimento

a) Correr las pruebas y recolectar datos

En las Tablas 4.5 y 4.6, se muestran los resultados promedio de 30 ejecuciones de los algoritmos *K-means*, *EC* y *N-means* para las instancias reales de 245,057, 414,528 y 657,308 objetos con valores de *k* =100, 200, 400. Sólo para *Skin* se presentan resultados con valores de *k*=800.

Tabla 4.5. Resultados promedio de 30 ejecuciones para instancias reales (tiempo)

Instancia	Grupos	Tiempo(ms) K-means	Tiempo(ms) EC	Tiempo(ms) N-means
Skin	100	307039.0	89579.0	86478.4
Paris	100	180303.8	67223.6	66190.5
New York	100	444680.0	177269.1	172636.5
Skin	200	677579.7	215769.5	194447.7
Paris	200	365715.7	165101.7	157618.9
New York	200	1160835.4	417023.4	396596.7
Skin	400	1507180.1	500606.5	400913.2
Paris	400	937485.1	488187.9	446912.4
New York	400	2007871.1	1019215.1	905649.5
Skin	800	2331976.1	1188085.1	837659.4

Tabla 4.6. Resultados promedio de 30 ejecuciones para instancias reales (calidad)

Instancia	Grupos	Calidad K-means	Calidad EC	Calidad N-means
Skin	100	1894067.9	1999685.3	2000066.6
Paris	100	1216.6	1255.5	1255.6
New York	100	3120.7	3302.2	3302.6
Skin	200	1367798.6	1444779.3	1446035.9
Paris	200	806.3	828.1	828.4
New York	200	2119.0	2240.1	2240.6
Skin	400	996499.0	1050332.8	1052154.1
Paris	400	536.8	548.4	549.0
New York	400	1441.8	1510.6	1511.8
Skin	800	726481.5	756357.8	759612.5

b) Análisis de los datos

En las columnas tres y cuatro de la Tabla 4.7, se muestra la diferencia del tiempo de ejecución para los algoritmos *EC* y *N-means* en contraste con el algoritmo *K-means*. En la última columna se muestra la diferencia de tiempo entre los algoritmos *EC* y *N-means*. Como se observa (ver valores sombreados), cuando se incrementa el número de grupos para cada una de las instancias dadas, *N-means* obtiene mejores resultados en comparación con el algoritmo EC.

Tabla 4.7. Diferencias de tiempo de ejecución entre los algoritmos K-means, EC y N-means.

Instancia	Grupos	Diferencia (%) K-Means Vs. EC	Diferencia (%) K-means Vs. N-means	Diferencia (%) EC Vs. N-means
Skin	100	70.824	71.834	3.46
Paris	100	62.716	63.289	1.53
New York	100	60.135	61.177	2.61
Skin	200	68.155	71.302	9.88
Paris	200	54.855	56.901	4.53
New York	200	64.075	65.835	4.89
Skin	400	66.785	73.399	19.91
Paris	400	47.925	52.328	8.45
New York	400	49.239	54.895	11.14
Skin	800	49.052	64.07	29.49

En las columnas tres y cuatro de la Tabla 4.8, se muestran las diferencias de calidad de agrupamiento entre los algoritmos *EC* y *N-means* en contraste con el algoritmo *K-means*. En la última columna se muestra la diferencia en la calidad entre los algoritmos *EC* y *N-means*. Como se puede observar (ver valores sombreados), la disminución en la calidad no fue significativa.

Tabla 4.8. Diferencias de calidad entre los algoritmos K-means, EC y N-means

Instancia	Grupos	Diferencia (%) K-Means Vs. EC	Diferencia (%) K-means Vs. N-means	Diferencia (%) EC Vs. N-means
Skin	100	-5.576	-5.596	-0.019
Paris	100	-3.203	-3.210	-0.006
New York	100	-5.815	-5.827	-0.011
Skin	200	-5.628	-5.719	-0.086
Paris	200	-2.702	-2.741	-0.038
New York	200	-5.716	-5.737	-0.019
Skin	400	-5.402	-5.585	-0.173
Paris	400	-2.156	-2.269	-0.109
New York	400	-4.774	-4.854	-0.076
Skin	800	-4.1124	-4.560	-0.430

En las Figuras 4.7, 4.8 y 4.9, se muestran gráficamente los resultados promedio de 30 ejecuciones del tiempo de ejecución para las instancias reales de 245,057, 414,528 y 657,308 objetos. Como se observa a medida que el número de grupos se incrementa, la complejidad del algoritmo *N-means* (línea negra) es menor en comparación con los algoritmos *EC* (línea azul) y *K-means* (línea roja) para todos los casos.

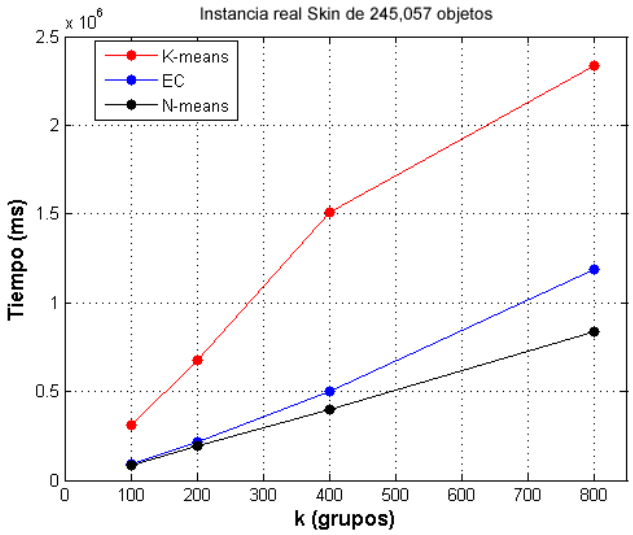

Figura 4.7. Comparación de tiempo de ejecución para una instancia real de 245,057 objetos.

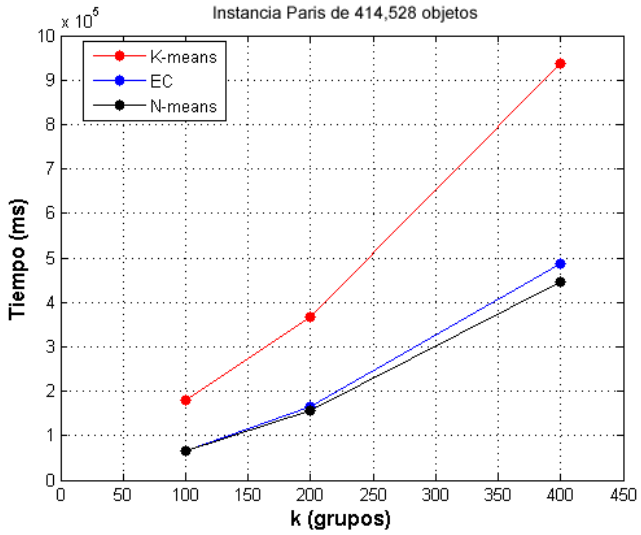

Figura 4.8. Comparación de tiempo de ejecución para una instancia real de 414,528 objetos.

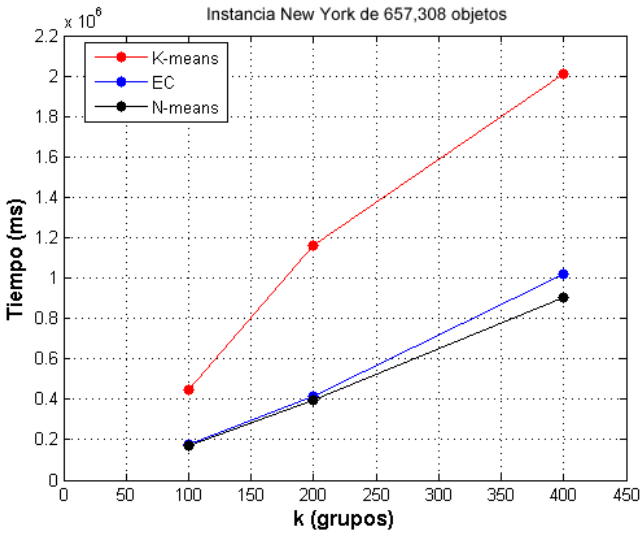

Figura 4.9. Comparación de tiempo de ejecución para una instancia real de 657,308 objetos.

En las Figuras 4.10, 4.11 y 4.12, se representan gráficamente los resultados promedio de 30 ejecuciones de la calidad de agrupamiento para las instancias reales de 245,057, 414,528 y 657,308 objetos. Se observa que la disminución de la calidad no fue significativa.

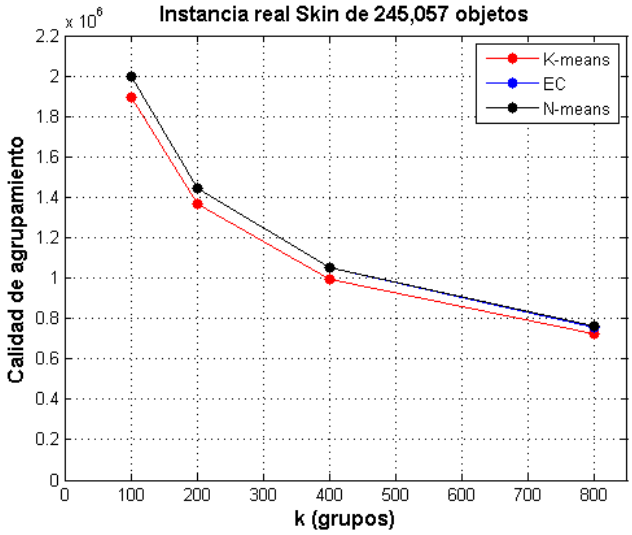

Figura 4.10. Comparación de calidad de agrupamiento para una instancia real de 245,057 objetos.

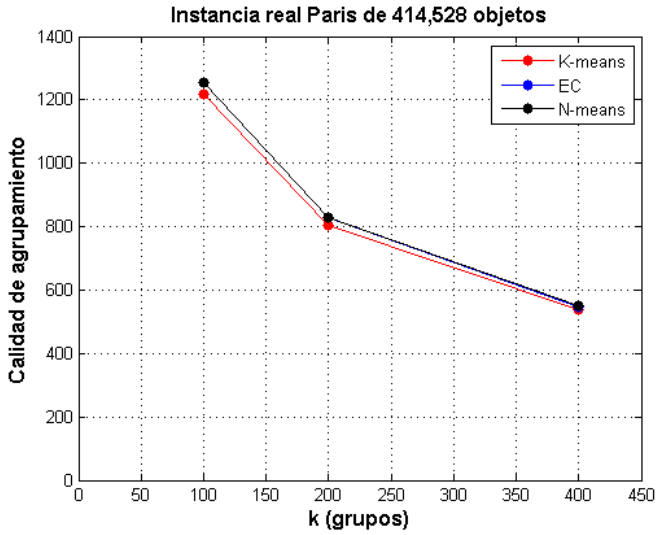

Figura 4.11. Comparación de calidad de agrupamiento para una instancia real de 414,528 objetos.

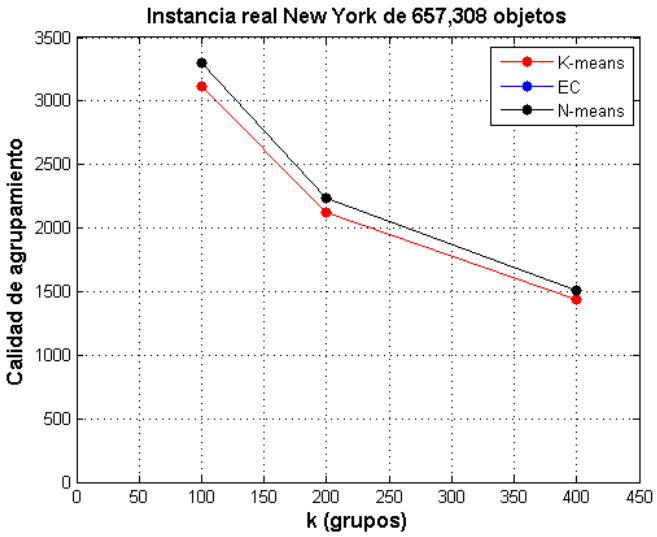

Figura 4.12. Comparación de calidad de agrupamiento para una instancia real de 657,308 objetos.

4.3. Resultados destacables para instancias sintéticas y reales

En el caso particular de una instancia sintética de 40,000 objetos, con un valor de *k*=800, *N-means* redujo el tiempo de ejecución en 75.5% y se disminuyó la calidad a solamente -1.52%. En la misma instancia aplicando *Early Classification*, el tiempo de ejecución se redujo en 67.7% y se disminuyó la calidad a -1.31%.

Con una instancia real de 245,057 objetos, con un valor de *k*=800, *N-means* redujo el tiempo de ejecución en un 64%, con una disminución de la calidad a sólo -4.56%. En la misma instancia real aplicando *Early Classification*, el tiempo de ejecución se redujo en 49% y una disminución de la calidad a -4.11%.

Es importante destacar que, con base en el análisis de los resultados tanto utilizando instancias sintéticas como instancias reales, se observó que al incrementar el número de grupos para cualquiera de las instancias, *N-means* es mejor en tiempo de respuesta que *EC* y presenta un comportamiento cuasi lineal.

Capítulo 5
Conclusiones y trabajos futuros

"La verdadera ciencia enseña, por encima de todo, a dudar y a ser ignorante"
Miguel de Unamuno (1864 - 1936)

En el presente capítulo se muestran las principales conclusiones derivadas del análisis experimental realizado para mostrar la viabilidad de la meta-heurística N-means (Sección 5.1). Asimismo, en la Sección 5.2 se proponen temas que permitirán desarrollar futuras investigaciones. Finalmente, en la Sección 5.3 se muestran las publicaciones que se originaron con relación a los avances y resultados de esta investigación.

5.1. Conclusiones

En esta investigación se muestra que es factible incrementar la eficiencia del algoritmo K-means mediante la mejora de la heurística *Early Classification*.

En particular la investigación se enfocó en el estudio, análisis y desarrollo de una meta-heurística denominada N-means. La finalidad del estudio fue incrementar la eficiencia de la heurística *Early Classification*. Con este objetivo se integraron las heurísticas: *Early Classification* y *grupos estables* y como resultado se logró desarrollar la meta-heurística N-means. A continuación se sintetizan los principales resultados del estudio.

Como producto de la observación de las ejecuciones del algoritmo K-means, se determinó que algunos grupos se estabilizan primero que otros. Por lo anterior se constituyen grupos estables, es decir, aquellos que ya no presentan intercambio de objetos con otros grupos en iteraciones posteriores.

Para validar N-means se utilizaron tres instancias sintéticas y tres reales. Los resultados obtenidos se contrastaron con los algoritmos K-means, *Early Classification* y N-means. De lo anterior se obtuvieron resultados relevantes, los cuales se destacan a continuación:

a) Para una instancia sintética de 40,000 objetos, con un valor de k=800, N-means redujo el tiempo de ejecución en 75.5% y se logró una disminución de la calidad de agrupamiento a sólo -1.52%. Esa misma instancia con *Early Classification*, el tiempo de ejecución se redujo en 67.7% y se logró una disminución de la calidad de agrupamiento a sólo -1.31%.
b) Para una instancia real de 245,057 objetos, con un valor de k=800, N-means redujo el tiempo de ejecución en 64% y se logró disminuir la calidad de agrupamiento a solamente -4.56%. En esa misma instancia con *Early Classification*, el tiempo de ejecución se redujo en 49% y se logró disminuir la calidad de agrupamiento a solamente -4.11%.

Con base en el análisis de los resultados que se presentaron en el Capítulo 4, se mostró que al incrementar el número de grupos para cualquier instancia dada, N-means presentó un comportamiento cuasi lineal. Con esto se muestra que el desempeño de N-means es mejor que *Early Classification*.

5.2. Trabajos futuros

Para desarrollar estudios futuros que den continuidad a la presente investigación, se propone:

a) Integrar a la meta-heurística N-means la idea que expone Fahim [15]. Con base en el estudio y análisis de dicha propuesta, se considera que integrándose con N-means es posible obtener resultados alentadores.
b) Realizar de manera experimental, un análisis comparativo de N-means con los algoritmos propuestos por Fahim [15] y Tsai [5]. Lo anterior requiere de implementar dichos algoritmos.
c) Experimentar con instancias sintéticas de manera visual aplicando el algoritmo K-means. De esta manera pueden surgir nuevas ideas para incrementar su eficiencia.

5.3. Publicaciones

Como productos de los avances logrados con la presente investigación, se publicaron los siguientes artículos:

- En la International Conference of Numerical Analysis and Applied Mathematics (ICNAAM) llevada a cabo en Grecia, 2014, se presentó el artículo *An Improvement to the K-means Algorithm Oriented to Big Data*.

- En el Encuentro Nacional de Ciencias de la Computación (ENC) llevado a cabo en Oaxaca, 2014, se presentó el artículo *Mejora del algoritmo K-means mediante una meta-heurística orientada a la reducción de su complejidad computacional.*

REFERENCIAS

[1] J. Macqueen, “Some methods for classification and analysis of multivariate observations,” *Proc. Fifth Berkeley Symp. Math. Stat. Probab.*, vol. 233, no. 233, pp. 281–297.

[2] J. Wu, *Advances in K-means Clustering*. Springer Theses Recognizing Outstanding Ph.D. Research, 2012, p. 187.

[3] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, *Top 10 algorithms in data mining*, vol. 14, no. 1. 2007, pp. 1–37.

[4] J. Pérez, M. F. Henriques, R. Pazos, L. Cruz, G. Reyes, J. Salinas, and A. Mexicano, “Mejora al algoritmo de agrupamiento K-Means mediante un nuevo criterio de convergencia y su aplicación a bases de datos poblacionales de cáncer,” *2do Taller Lat. Iberoam. Investig. Operaciones.*, pp. 1–7, 2007.

[5] M.-C. Chiang, C.-W. Tsai, and C.-S. Yang, “A time-efficient pattern reduction algorithm for k-means clustering,” *Inf. Sci. (Ny).*, vol. 181, no. 4, pp. 716–731, 2011.

[6] Jim Z.C. Lai and Y.-C. Liaw, “Improvement of the k-means clustering filtering algorithm,” *ELSEVIER, Pattern Recognit.*, vol. 41, pp. 3677–3681, 2008.

[7] J. Pérez, C. E. Pires, L. Balby, A. Mexicano, and M. Á. Hidalgo, “Early Classification : A New Heuristic to Improve the Classification Step of K-Means,” *JIDM - J. Inf. Data Manag.*, vol. 4, no. 2, pp. 94–103, 2013.

[8] M. A. Barrón, “Desarrollo de un prototipo para la aplicación de técnicas de minería de datos sobre una base de datos real de base poblacional de cáncer,” Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, México, 2008.

[9] M. del R. Boone, “Identificación de Regiones con Altas Tasas de Incidencia de Cáncer mediante la Integración y Uso de Técnicas de Minería de Datos: Almacenes de Datos, Agrupamiento y Sistemas de información Geográficos,” Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, México, 2011.

[10] A. Mexicano, “Caracterización de conjuntos de instancias difíciles del problema de Bin Packing orientada a la mejora de algoritmos metaheurísticos mediante el uso de técnicas de minería de datos,” Tesis Doctoral, Centro Nacional de Investigación y Desarrollo Tecnológico, México, 2012.

[11] R. I. Basave, “Mejoramiento de la Eficiencia y Eficacia del algoritmo de agrupamiento K-Means mediante una nueva Condición de Convergencia,” Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, México, 2005.

[12] A. Moreno, “Mejora del Algoritmo K-Means Incrementando su Eficiencia en la Fase de Clasificación,” Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, México, 2013.

[13] J. Pérez, A. Mexicano, R. Pazos, R. Santaolaya, M. Hidalgo, A. Moreno, and N. Almanza, “Improvement to the K-Means Algorithm Through a Heuristics Based on a Bee Honeycomb Structure,” *J. Netw. Innov. Comput.*, vol. 1, pp. 119–125, 2013.

[14] J. Z. C. Lai, T.-J. Huang, and Y.-C. Liaw, “A fast -means clustering algorithm using cluster center displacement,” *Pattern Recognit.*, vol. 42, no. 11, pp. 2551–2556, 2009.

[15] A. Fahim, A. Salem, F. Torkey, and M. Ramadan, “An efficient enhanced k-means clustering algorithm,” *J. Zhejiang Univ. Sci. A*, vol. 7, no. 10, pp. 1626–1633, 2006.

[16] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[17] S. J. Nanda and G. Panda, “A survey on nature inspired metaheuristic algorithms for partitional clustering,” *Swarm Evol. Comput.*, vol. 16, pp. 1–18, 2014.

[18] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–78, 2005.

[19] L. Morissette and S. Chartier, “The k-means clustering technique : General considerations and implementation in Mathematica,” *Tutor. Quant. Methods Psychol.*, vol. 9, no. 1, pp. 15–24, 2013.

[20] H. Bock, “Origins and extensions of the k -means algorithm in cluster analysis,” vol. 4, no. December, pp. 1–18, 2008.

[21] M. E. Celebi, H. a. Kingravi, and P. a. Vela, “A comparative study of efficient initialization methods for the k-means clustering algorithm,” *Expert Syst. Appl.*, vol. 40, no. 1, pp. 200–210, 2013.

[22] M. Erisoglu, N. Calis, and S. Sakallioglu, “A new algorithm for initial cluster centers in k-means algorithm,” *Pattern Recognit. Lett.*, vol. 32, no. 14, pp. 1701–1705, 2011.

[23] D. Reddy and P. K. Jana, “Initialization for K-means Clustering using Voronoi Diagram,” *Procedia Technol.*, vol. 4, pp. 395–400, 2012.

[24] J. Pérez, R. Pazos, L. Cruz, G. Reyes, R. Basave, and Héctor Fraire, “Improving the Efficiency and Efficacy of the K-means Clustering Algorithm Through a New Convergence Condition,” *Springer-Verlag Berlin Heidelb.*, pp. 674–682, 2007.

[25] S.-S. LEE and J.-C. LIN, “An accelerated K-means clustering algorithm using selection and erasure rules,” *J. Zhejiang Univ. Sci. C*, vol. 13, no. 10, pp. 761–768, 2012.

[26] C. C. Mcgeoch, *A Guide to Experimental Algorithmics*. Cambridge University Press, NY, USA, 2012.