RISI **2**(2), 16-25 (2005)
Rev. investig. sist. inform.
Facultad de Ingeniería de Sistemas e Informática
Universidad Nacional Mayor de San Marcos
ISSN: 1815-0268 (versión impresa)/ ISSN: 1816-3823 (versión electrónica)

GRASP EN LA RESOLUCIÓN DEL PROBLEMA DE CLUSTERING

GRASP IN THE RESOLUTION OF THE CLUSTERING PROBLEM

Erick Vicente*, Luis Rivera**, David Mauricio***

RESUMEN El clustering puede ser abordado como un problema de optimización combinatoria cuando los clusters son una partición de un conjunto de objetos. La meta heurística Grasp es una técnica relativamente reciente que ha sido utilizada para resolver de manera eficiente múltiples problemas de optimización combinatoria. En este trabajo, adaptamos la meta heurística Grasp para la resolución del problema del clustering basado en los principios del algoritmo K-Means. El algoritmo propuesto, denominado GraspKM, aprovecha la rápida convergencia del algoritmo K-Means evitando el inconveniente de alcanzar óptimos locales. El algoritmo demuestra ser superior al algoritmo K-Means y es comparable con otras meta heurísticas revisadas en cuanto a eficiencia. Los experimentos computacionales han sido realizados con colecciones de datos ampliamente usados en la literatura sobre clustering.

Palabras clave: Grasp, K-Means, Clustering, Clasificación, Meta Heurística.

ABSTRACT The clustering could be approached as a combinatorial optimization problem when the clusters are a partition of an objects set. The Grasp meta-heuristic is a relatively recent technic that had been used to solve of an e_cient manner several combinatorial optimization problems. In this work, we adapted the Grasp metaheuristic to solve the clustering problem based on the basis of K-Means algorithm. The proposed algorithm, named GraspKM, takes advantages of fast convergence of K-Means algorithm avoiding the inconvenience of obtaining a local optimal. The algorithm shows to be better than K-Means algorithm and it is comparable with another meta-heuristic method reviewed with respect to e_ciency. The computational experiments had been realized with a data collection extensively used on clustering literature.

Keywords: Grasp, K-Means, Clustering, Classification, Meta Heuristic.

1. INTRODUCCIÓN

Dado un conjunto de objetos, el proceso de clustering debe encontrar grupos cuyos elementos sean similares entre sí, y a la vez diferentes a los elementos de los otros grupos. Los grupos con esas características son conocidos como clusters. Los objetos son representados por D atributos descriptores en forma de vectores en el espacio RD, y con una medida de comparación de la símil arridad, como la distancia, se conformarían los clusters con objetos similares. En el proceso de la conformación de los grupos, que en adelante será conocido como clustering, no existe conocimiento previo acerca de cómo se debe conformar un cluster; por tal motivo, el proceso de clustering es

* Universidad Nacional Mayor de San Marcos, Unidad de Postgrado FISI, Lima-Perú
** Universidad Federal Norte Fulmínense, LCMAT-CCT, Río de Janeiro-Brasil
*** Universidad Nacional Mayor de San Marcos, Unidad de Postgrado FISI, Universidad Ricardo Palma, Facultad de Ingeniería, Lima-Perú
E-mail: erick.vicente@gmail.com, rivera@uenf.br , research@yahoo.com

también conocido como clasificación no supervisada. Los tipos de objetos varían de acuerdo con el contexto de la aplicación del clustering; por ejemplo, en las tareas de clasificación dentro de la minería de datos, los objetos serán registros de la base de datos; en la recuperación de la información los objetos serían documentos; y en procesamiento de imágenes los objetos serán los píxeles que conforman la imagen.

El clustering tiene múltiples aplicaciones dentro de las ciencias de la computación, como compresión de imágenes [25] y voz digitalizadas [17]; en la recuperación de informaciones relacionadas [2]; en minería de datos, donde se buscan grupos con ciertas características de interés (por ejemplo, descubrimiento de nuevos segmentos de clientes con el fin de mejorar los servicios que brinda una determinada empresa) [8]; en la segmentación de imágenes para dividir la imagen en regiones homogéneas (según alguna característica de interés como la intensidad, color o textura) en aplicaciones médicas [24], clasificación de imágenes satelitales en zonas (urbana, descampados, bosques, ríos) [26].

Los métodos de clustering existentes difieren uno del otro en la forma de estructurar los clusters. Aquellos que encuentran clusters que corresponden a una partición del conjunto de objetos se les conoce como métodos de Hard-clustering [16] o clustering particional [13], siendo el más conocido el algoritmo K-Means [10, 18].

A los métodos que asignan a cada objeto un valor de pertenencia con respecto a cada cluster se les conoce como métodos de Soft-clustering [16], y entre los más representativos de este tipo de clustering se encuentran los algoritmos Fuzzy C-Means [4] y Expectación Maximización [7]. El método propuesto en el presente trabajo se considera dentro de las técnicas de Hard-clustering o clustering particional; por tanto, se definirá el problema del clustering desde ese punto de vista.

PROBLEMA DE CLUSTERING

Dado un conjunto de n objetos denotado por $X = \{x_1, x_2, ..., x_n\}$, en que $x_i \in R^D$, sea K un número entero positivo conocido a priori, el problema del clustering consiste en encontrar una partición:

$P = \{C_1, C_2, ..., C_K\}$ de X, siendo C_j un cluster conformado por objetos similares, satisfaciendo una función objetivo $f : R^D \rightarrow R$, y las condiciones:

$$C_i \cap C_j = \varnothing \text{ para } i \neq j, \text{ y } \cup C_i = X$$

Para medir la símil arridad entre dos objetos xa y x_b se usará una función de distancia denotada por $d(x_a, x_b)$, siendo la distancia euclidiana la más usada para medir la símil arridad. Así la distancia entre dos diferentes elementos $x_i = (x_{i1}, ..., x_{iD})$ y $x_j = (x_{j1}, ..., x_{jD})$ es $d(x_i, x_j) = \sqrt{\sum_{l=1}^{D} (X_{il} - X_{jl})^2}$ Los objetos de un cluster son similares cuando las distancias entre ellos es mínima; esto permite formular la función objetivo f, como:

$$\sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, \bar{x}_j)^2; \quad (1)$$

esto es, se desea minimizar (1); donde $\bar{x}_j$, conocido como elemento representativo del cluster, es la media de los elementos del cluster Cj ,

$$\underline{x}_j = \frac{1}{|C_j|} \sum x_i \in C_j, \quad (2)$$

y corresponde al centro del cluster. Bajo esas características, el clustering es un problema de optimización combinatoria, y ha sido demostrado que es un NP-Difícil [5].

MÉTODO PROPUESTO

El algoritmo K-Means es una técnica clustering ampliamente usada y capaz de encontrar rápidamente una solución minimizando (1). Dentro de los principales inconvenientes del algoritmo K-Means está su alta dependencia de la elección de los centros iniciales y su convergencia a óptimos locales. Esta última (2) deficiencia es en realidad una debilidad de los algoritmos golosos, que son rápidos encontrando soluciones pero quedan atrapados en óptimos locales. Feo y Resende [9] proponen una meta heurística llamada Grasp (Greedy Randomised Adaptative Search Procedure) que aprovecha la efectividad de los algoritmos golosos adaptando la forma voraz de construir las soluciones para evitar la convergencia a óptimos locales y luego mejorar las soluciones encontradas. El presente trabajo tiene como objetivo la adaptación del algoritmo K-Means dentro del procedimiento Grasp para la obtención de mejores soluciones que el algoritmo K-Means y comparables a las soluciones obtenidas con otras meta heurísticas propuestas para el problema del clustering. Esto es, respecto a inestabilidad de los resultados y robustez del método.

Para alcanzar el objetivo propuesto en este trabajo, en la Sección 2 se hace una breve revisión de los

métodos de clustering. En la Sección 3 se presenta el algoritmo GraspKM para el problema del clustering.

En la Sección 4 se describen las colecciones de datos usadas para la prueba del algoritmo, así como el análisis de los resultados obtenidos. Finalmente, en la Sección 5 se exponen las conclusiones y trabajos futuros.

2. MÉTODOS DE CLUSTERING

Los métodos expuestos se encuentran dentro de lo que se ha clasificado como hard clustering, y ofrecen soluciones subóptimas para el problema del clustering. En la primera parte se aborda el algoritmo K-Means, al cual se le da una especial atención debido a que su adaptación dentro del marco de la meta heurística Grasp es parte principal del presente trabajo. Luego, se hace una revisión de las meta heurísticas para el problema del clustering propuestas recientemente, dentro de las que podemos encontrar los algoritmos genéticos, los algoritmos eméticos y la meta heurística Grasp.

2.1. Algoritmo K-MEANS

El algoritmo K-Means es una de las heurísticas comúnmente utilizadas para resolver el problema de clustering [18, 10]. La idea básica del algoritmo es obtener los K centros iniciales y formar clusters asociando todos los objetos de X a los centros más cercanos, después se recalculan los centros. Si esos centros no difieren de los centros anteriores, entonces el algoritmo termina; caso contrario, se repite el proceso de asociación con los nuevos centros hasta que no haya variación en los centros, o se cumpla algún otro criterio de parada como poco número de reasignaciones de los objetos.

Los K diferentes centros iniciales $\{\overline{x}_j\}_{j=1,\ldots,K}$ se seleccionan aleatoriamente de X. La asociación del objeto $x_i \in X$ con el centro más cercano $\overline{x}_j$ del cluster C_j es dada si $d(x_i, \overline{x}_j) < d(x_i, \overline{x}_p)$ para todo j, p = 1, ..., K y $j \neq p$. Los centros son recalculados usando la expresión (2). La idea del algoritmo K-Means se presenta en seudo-código, como:

Algoritmo K-Means ($X = \{x_1, ..., x_n\}$, K)

1. Seleccionar aleatoriamente de X centros iniciales $\{\overline{x}_i\}_{i=1,\ldots,K}$
2. Para cada $x_i \in X$

2.1. Asociar x_i con el centro más cercano:

$C_j = C_j \cup \{x_i\}$, si $d(x_i, \overline{x}_j) < d(x_i, \overline{x}_p)$, $\forall$ j, p = 1, ..., K y $j \neq p$

3. Fin para

4. Calcular los centros $\overline{x}^*_i = \frac{1}{|C_i|}\sum_{j=1}^{|C_i|} xj$, para $x_j \in C_i$

5. Si no hay más reasignaciones: $\overline{x}^*_i = \overline{x}_i$, $\forall$ i, parar.

Caso contrario, considerar $\overline{x}^*_i$ como nuevo centro $\overline{x}_i$, e ir al paso 2.

Los principales inconvenientes del algoritmo K-Means citados por Peña *et al.* [23] son: su sensibilidad a la inicialización (debido a esto, el resultado final depende del estado inicial); el K-Means es un algoritmo de búsqueda local (el algoritmo minimiza la función objetivo dada en (1), pero no garantiza una configuración óptima de los clusters); y se debe tener conocimiento previo del valor de K. Peña *et al.* [23], quienes discuten cuatro métodos para la inicialización del algoritmo K-Means. El primer método de inicialización es completamente aleatorio; el segundo es el método de Forgy [10]; el tercero es el método de McQueen [18]; y finalmente, el método de Kauman y Rousseeuw [15]. Los tres primeros métodos son, de alguna manera, aleatorios; sólo el algoritmo propuesto por Kauman y Rouseeuw es un algoritmo heurístico que identifica los objetos más representativos que prometen tener en su alrededor gran cantidad de objetos. Peña *et al.* concluyen que la inicialización completamente aleatoria y la propuesta de Kauman y Rouseeuw ofrecen una mejor inicialización para el algoritmo K-Means que el resto de métodos, haciéndolo más robusto.

2.2. Métodos meta heurísticos

Los algoritmos genéticos han sido propuestos para el hard clustering por Murthy y Chowdhury [20], Bandyopadhyay y Maulik [1], y Pacheco y Valencia [21]. En [20] se codifican las soluciones en cromosomas de longitud igual al número de elementos de X, por lo que el método es limitado por el número de elementos de X. Las otras propuestas codifican los cromosomas con los valores de los centros de los clusters. En estos casos, un cromosoma está compuesto por un vector $a = (\overline{x}_1, \ldots, \overline{x}_K)$, donde $\overline{x}_i$ es centro de C_i; la implementación requiere de menos recursos, y el método es factible para cualquier número de elementos de X. Es decir, las propuestas [1] y [21] son parecidas en el aspecto de codificación de cromosomas,

pero varían en cuanto a los operadores de cruzamiento y mutación, y que el primero utiliza el algoritmo K-Means para refinar la solución en cada generación de la población.

Los algoritmos meméticos han sido propuestos para el hard clustering por Pacheco y Valencia [21] y Merz [19]. Al igual que los algoritmos genéticos, los algoritmos meméticos, utilizan poblaciones de soluciones denominadas memes, que se van recombinando generación tras generación en búsqueda de un óptimo. La diferencia radica en que cada meme es obtenido por un algoritmo de búsqueda local en un espacio de soluciones de óptimos locales. En [19] se propone el uso del algoritmo K-Means para la generación de los óptimo locales, mientras que en [21] se realizan experimentos con diversos algoritmos de búsqueda local, tales como HK-Means y J-Means [12]. Tanto en [19], como en [21] se codifican los memes con los centros de los clusters; es decir, un meme es compuesto por un vector $a = (\bar{x}_1, ..., \bar{x}_K)$ de centros obtenidos con un algoritmo de búsqueda local.

3. META HEURÍSTICA GRASP PARA EL PROBLEMA DEL CLUSTERING

En parte, algunas inconveniencias del algoritmo K-Means fueron superadas con la ayuda de novedosas técnicas dentro de la optimización combinatoria, tales como algoritmos genéticos, algoritmos meméticos y Grasp. Pero aún continúan las inconveniencias, incluyéndose robustez de la convergencia a la solución e inicialización.

En esa perspectiva, proponemos una adaptación del algoritmo K-Means para la obtención de los centros iniciales, los cuales son alterados si no cumplen un criterio de evaluación establecido, y son refinados mediante un proceso de búsqueda local. Los procesos están diseñados dentro del marco de la meta heurística Grasp.

3.1. Meta heurística GRASP

Un procedimiento de búsqueda voraz, aleatoria y adaptativa (GRASP) es una meta heurística propuesta por Feo y Resende [9] para encontrar soluciones aproximadas de problemas de optimización combinatoria, mediante un proceso iterativo. En cada iteración se realizan dos fases de operaciones: construcción y búsqueda local. En la fase de construcción se genera un conjunto solución **S** de una instancia **E** de un problema combinatorio, y en la fase de búsqueda local se determina una posible mejor solución a **S**; finalmente, se elige la solución mejor entre la solución de la iteración anterior y la actual. La mejor solución será indicada por una función objetivo f. Cada iteración es realizada un número máximo de veces (MAX_ITER). A continuación se presenta en notación de pseudocódigo o algoritmo GRASP básico tal como fue descrito:

Algoritmo Grasp (E, MAX_ITER, α)

1. Inicializar solución S : = $\varnothing$ y f* := ∞
2. Repetir MAX_ITER veces
2.1. Obtener una solución S* de Construcción_Grasp (E, α)
2.2. Obtener una solución S* de Búsqueda_Local_ Grasp (S*)
2.3. Si f(S*) < f*, entonces
2.3.1. Actualizar S:= S* y f* := f(S*)
2.4. Fin Si
3. Fin Repetir
4. Solución S

El hecho de que la fase de búsqueda local toma como entrada la solución obtenida en la fase de construcción proporciona un conocimiento frente a los algoritmos de búsqueda local tradicionales.

Construcción Grasp

En esta fase se construye un conjunto de soluciones con base en la adaptación de un algoritmo goloso. Tales algoritmos tienen una función de evaluación golosa g: C$\rightarrow$ R que selecciona el mejor elemento de un conjunto de candidatos C a ser incorporados en la solución. El criterio de selección goloso de g depende del carácter del problema, puede ser maximización o minimización. El constructor de soluciones evita el determinismo de los algoritmos golosos, utilizando un parámetro de relajación _ para formar una lista restringida de candidatos (Restricted Candidate List - RCL) alrededor del mejor elemento a seleccionar. El elemento a ser incorporado en la solución es elegido aleatoriamente del RCL. Esta forma de selección proporciona al Grasp un aspecto estocástico de selección con tendencia a los mejores elementos que permite evitar los óptimos locales. El parámetro de relajación $\alpha \in$ [0, 1] indica la amplitud del RCL alrededor del mejor candidato.

Cuando α = 0, el RCL estará conformado sólo por el mejor candidato y la fase de construcción se comportará como un algoritmo goloso. Cuando α = 1, el RCL estará conformado por el total de elementos de C y la selección de los candidatos será totalmente aleatoria. El mejor valor de α para el problema en estudio se obtiene a través de múltiples experimentos computacionales de calibración.

Búsqueda local Grasp

La búsqueda local se realiza de manera iterativa, explorando en la vecindad de un conjunto de solución S generada por la operación de construcción. El desempeño de la operación de búsqueda local dependerá del método elegido. Si N es una vecindad de soluciones, se dice que S' $\in$ N(S) es un óptimo local si f(S') < f(S). No existe un esquema de búsqueda local específico a utilizarse, sólo es necesario que mejore la solución encontrada en la fase de construcción.

3.2. Grasp basado en K-MEANS

Adaptamos la meta heurística Grasp para resolver de manera eficiente el problema del clustering minimizando la función objetivo dada en (1). En ese sentido, enfocamos el algoritmo K-Means considerando las dos fases de la arquitectura de Grasp y una fase adicional previa a estas, llamada inicialización KM. El algoritmo formulado debe realizar repetidas veces (MAX_ ITER veces) la secuencia de las tres fases mencionadas. Así, en la fase de inicialización (InicializaciónKM) se obtienen los K centros iniciales y se definen los clusters C_j en torno de sus centros iniciales. Estos clusters sirven de base para la fase de construcción (ConstrucciónKM) donde se refina la solución inicial, evitando caer en óptimos locales. La siguiente fase, búsqueda de la mejor solución (MemoriaKM), se basa en la exploración de nuevas soluciones alterando heurísticamente la estructura de los clusters obtenidos en la fase de construcción para mejorar la solución. Por último, retiene la mejor solución entre la anterior y la actual. Presentamos la estructura del algoritmo GraspKM, para después presentar en detalle cada una de las fases mencionadas. Se consideran como datos de entrada el conjunto de objetos X, un número K de clusters a generar, la relajación α, y el máximo número de iteraciones MAX_ITER. Debemos esperar como resultado el conjunto de clusters C.

Algoritmo GraspKM (X, K, MAX_ITER)

1. f* := ∞, C := {}
2. Repetir MAX_ITER veces
2.1. C' := InicializacionKM(X, K)
2.2. C' := ConstruccionKM(X, K, C', α_)
2.3. C' := MemoriaKM(X, K, C')
2.4. Si f(C') < f*, entonces
2.4.1. C:= C'
2.4.2. f*:= f(C')
2.5. Fin Si
3. Fin Repetir
4. Solución C

3.2.1. Configuración inicial

De manera similar al algoritmo K-Means, en esta fase se seleccionan K centros aleatoriamente, luego se forman los clusters iniciales asociando el objeto x $\in$ X al cluster C_j si el centro $\bar{x}_j$ es el menos distante al objeto. Finalmente, se calcula el nuevo centro o la media del cluster C_j, (j = 1, ..., K) haciendo uso de la expresión (2). Seguidamente, presentamos el algoritmo InicializacionKM:

InicializacionKM (X,K)

1. Seleccionar K centros iniciales $\{\bar{x}_i = \text{Random}(X)\}_{i=1,...,K}$
2. Para cada x $\in$ X,
2.1. Asignar x a C_j, cuando j = ArgMin$\{d(x, \bar{x}_i)\}_{i=1,...,K}$
3. Fin Para
4. Calcular los centros $\{\bar{x}_j := \text{Media}(C_j)\}_{j=1,..., K}$
5. Resultado C = $\{C_j\}_{j=1,...,K}$

3.2.2. Construcción de soluciones

Esta fase es una adaptación del algoritmo K-Means (Sección 2.1) con el objetivo de evitar la convergencia a óptimos locales. La adaptación se realiza principalmente sobre la función golosa que asigna los objetos (paso 2.1 del algoritmo K-Means), la cual establece que un objeto x $\in$ X se asigna al cluster C_j, si $\bar{x}_j$ es el centro menos distante al objeto x. Al aplicar el parámetro de relajación sobre la función golosa, se crea un conjunto RCL de posibles clusters a los cuales puede ser reasignado un objeto. El RCL estará conformado por una vecindad alrededor del cluster más próximo al objeto evaluado. Esta fase se implementa mediante el algoritmo Construcción KM que se presenta a seguir, considerando como datos de entrada el número de clusters K, el conjunto de clusters C = $\{C_j\}_{j=1,..., K}$ con sus respectivos centros $\{\bar{x}_i\}_{i=1,..., K}$ generados en la fase anterior, y el parámetro de relajación α.

ConstruccionKM (X,K,C, α)

1. Repetir

1.1. Para cada $x \in X$ tal que $x \in C_j$ para algún j = 1, ..., K

1.1.1. $\overline{\beta} := \text{Max}\{d(x, \overline{x}_l): d(x, \overline{x}_l) \leq d(x, \overline{x}_j)\}_{l=1,...,K}$

1.1.2. $\beta := \text{Min}\{d(x, \overline{x}_l)\}_{l=1,...,K}$

1.1.3. $RCL := \{C_t : d(x, \overline{x}_t) \leq \beta+\alpha(\overline{\beta}+\beta)\}_{t=1,...,K}$

1.1.4. $C_t := \text{Random}(RCL)$

1.1.5. Si $t \neq j$

$C_t := C_t \cup \{x\}$

$C_j := C_j - \{x\}$

1.1.6. Fin de Si

1.2. Fin de Para

1.3. Recalcular centros $\{\overline{x}_j := \text{Media}(C_j)\}_{j=1,...,K}$ Media(Cj)$\}_{j=1,...,K}$

2. Hasta que no haya más reasignaciones

3. Resultado $C = \{C_j\}_{j=1,...,K}$

En un proceso K-Means un objeto $x \in C_j$ será asignado a otro cluster C_p si la distancia $d(x, \overline{x}_p)$ es la mínima entre las distancias hacia los clusters y $j \neq p$. En el proceso Construcción KM, los posibles clusters que contendrían al objeto x en análisis son agrupados en un conjunto RCL que contiene un número menor de clusters cuyas distancias de sus centros al objeto x están en un intervalo definido por $\overline{\beta}$, que es el valor máximo de las distancias menores que la distancia a su centro de origen, β que es el mínimo de las distancias menores que la distancia a su centro de origen, regulada linealmente por el parámetro de relajación. Del conjunto RCL será elegido aleatoriamente un cluster al cual será reasignado el objeto x, desde luego retirándolo del cluster al cual correspondía antes de ese proceso. En los pasos que corresponden al segmento 1.1.4. del algoritmo ConstruccionKM son realizados las operaciones descritas. Esa operación de reasignación será realizada iterativamente para cada objeto $x \in X$. Después de la reasignación de todos los objetos de X en los diferentes clusters, es lógico que el centro haya variado; lo que justifica que, nuevamente, se deban recalcular los centros de cada cluster a través de la media aritmética de los objetos de los respectivos clusters, $\{\overline{x}_j := \text{Media}(C_j)\}_{j=1,...,K}$, donde la función Media es definida por (2). Propuesto de esta manera, la fase de construcción evita el determinismo del algoritmo K-Means, teniendo una gran probabilidad de encontrar una mejor solución, aunque también puede encontrar peores. Es por ese motivo que en el procedimiento Grasp, la fase de construcción debe procesarse repetidas veces para obtener una gran variedad de soluciones que pueden ser mejoradas en la fase de búsqueda local.

3.2.3. Búsqueda de la mejor solución

La fase de construcción de soluciones genera soluciones que no son necesariamente óptimas [9], debido a que la búsqueda se realiza de manera aleatoria en un espacio de soluciones restringido por el RCL. En la fase de búsqueda de la mejor solución, denominada Mejoría KM, se debe alterar la estructura de la solución generada en la fase de construcción con el fin de obtener una mejor solución.

Fränti y Kivijärvi [11] proponen un método de búsqueda local aleatorio para obtener una solución al problema del clustering. El método está basado en un proceso de búsqueda local que altera la estructura de la solución generando un centro aleatorio, seleccionado del conjunto de objetos X, y elimina el cluster con menor error cuadrático; luego, se reasignan los objetos a los clusters que tengan el centro más cercano; y, finalmente, se realiza un proceso de refinamiento de la solución mediante el algoritmo K-Means. El proceso es repetido un número determinado de veces, y el mejor resultado es devuelto como la solución del problema.

El método favorece la eliminación de clusters con menor error cuadrático que probablemente sea producto de alcanzar un óptimo local y evita este inconveniente a través de la generación de un nuevo centro. Basados en la propuesta de Fränti y Kivijärvi diseñamos la fase MejoriaKM. La idea básica es, dada una solución, ignorar y regenerar clusters según ciertos criterios establecidos. A diferencia de Fränti y Kivijärvi, nuestra propuesta elimina el cluster que contiene menos objetos y genera un nuevo centro aleatoriamente dentro del cluster más disperso. Luego, todos los objetos de X son reasignados a los clusters más cercanos; de esta manera, los objetos del cluster eliminado son repartidos entre el resto de clusters y, a la vez, un cluster se forma alrededor del nuevo centro. La configuración de los cluster obtenida hasta este punto no es la óptima, por lo cual la solución entra en un proceso de refinamiento, que para el caso sería el mismo de la fase de construcción Grasp. Todo el proceso es repetido iterativamente hasta que no se pueda encontrar otra mejor solución. La forma heurística con que se modifica la estructura de la solución, obedece a la posibilidad de encontrar una mejoría, asumiendo que los clusters con menor cantidad de elementos y mayor dispersión ocurren porque el algoritmo alcanzó un óptimo local y que se puede encontrar una mejor solución. En la mejoría descrita, podemos notar dos procesos que se pueden realizar de manera independiente. El primero de ellos es la alteración de la solución; es decir, la eliminación de un cluster, la generación de un nuevo cluster a partir de un centro elegido de manera aleatoria y la agrupación de los objetos alrededor

del nuevo centro y de los existentes. En adelante a este proceso se le denominará de reagrupación. El segundo es el proceso de refinamiento que corresponde al mismo de ConstruccionKM, lo que da un valor agregado a esta fase en cuanto a evitar alcanzar óptimos locales.

La estructura general del algoritmo MejoriaKM, se presenta a continuación. Considera como datos de entrada conjunto de datos X, el número de clusters K y los clusters generados por la fase construcción $C = \{C_j\}^{j=1, ..., K,}$ con respectivos centros $\bar{x}_j = \{\bar{x}_i\}_{i=1, ..., K}$. El proceso entra en una iteración hasta que se alcance una solución estable, realizando las dos etapas descritas anteriormente: la de reagrupación de los elementos de clusters críticos denominada ReagrupacionKM y el proceso ConstruccionKM usado para el refinamiento de la solución.

MejoriaKM (X,K,C')
1. Repetir
1.1. C := C'
1.2. C' := ReagrupacionKM(X,K,C)
1.3. C' := ConstruccionKM(X,K,C', α)
2. Mientras (f (C') < f(C))
3. Resultado $C = \{C_j\}_{j=1,...,K}$

Reagrupación

El proceso de Reagrupación KM requiere la identificación de los clusters C_l de menor número de elementos y C_h de mayor dispersión, y un nuevo centro $\bar{x}_r$, elegido de manera aleatoria de los elementos pertenecientes a C_h. Si C_l es el cluster con menor número de elementos, entonces l está dado por:

$$l = \text{ArgMin}\{|C_j|\}_{j=1,...,K} \quad (3)$$

Para la identificación del cluster de mayor dispersión, se necesita primero el cálculo del error promedio del cluster, el cual está dado por:

$$Ej = \frac{\sum x \in C_j d(x, \bar{x}_j)}{|C_j|} \quad (4)$$

Antes de identificar el cluster más disperso, se describirán dos casos extremos: el primero de un cluster compacto y el segundo de un cluster disperso. El primer caso es un cluster con error promedio bajo y que tiene una buena cantidad de objetos; esta situación nos da la idea de que el cluster está bastante compacto.

Por el contrario, si tenemos un cluster con error promedio alto y con gran cantidad de elementos, entonces diremos que el cluster está disperso. Es decir, cuanto mayor sea la relación E_j / $|C_j|$, mayor será la dispersión del cluster, por el contrario, menor será la dispersión y, por consiguiente, mayor será su compactación. La Figura 1 muestra la idea de un cluster compacto y otro disperso. Los clusters dispersos son los que nos interesa reagrupar, por ello se generará un nuevo centro dentro del cluster más disperso con la finalidad de que contribuya a una mejor distribución de los objetos entre los clusters.

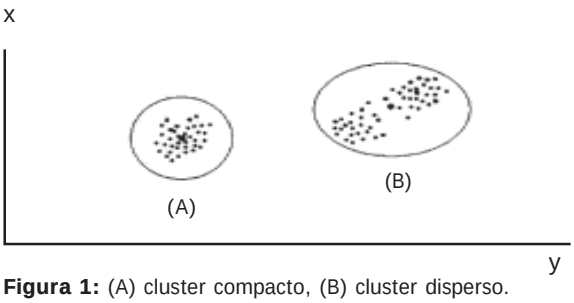

Figura 1: (A) cluster compacto, (B) cluster disperso.

Entonces, si C_h es el cluster con mayor dispersión, h está determinado por:

$$h = \text{ArgMax}\left\{\frac{Ej}{|C_j|}\right\}_{j=1,...,K, j \neq l} \quad (5)$$

El valor del nuevo centro $\bar{x}_r$ es tomado desde C_h cuya selección se realiza de manera aleatoria y esta dado por $\bar{x}_r = \text{Random}(C_h)$.

Los cálculos descritos anteriormente se llevan a cabo dentro del algoritmo ReagrupacionKM, cuyo pseudocódigo se presenta a continuación, donde los datos de entrada son los mismos de MejoriaKM:

ReagrupacionKM (X, K, C)

1. l := $\text{ArgMin}\{|C_j|\}_{j=1, ..., K}$
2. h := $\text{ArgMax}\left\{\frac{Ej}{|Cj|}\right\}_{j=1, ..., K, j \neq l}$
3. $\bar{x}_r$:= Random(C_h)
4. Reemplazar $\bar{x}_l$ por $\bar{x}_r$
5. Para cada x ∈ X // Generando clusters
5.1. Asignar x a C_j, donde $j = \text{ArgMin}\{d(x,x_i)\}_{i=1, ..., K}$
6. Fin Para
7. Calcular los centros $\{\bar{x}_j := \text{Media}(C_j)\}_{j=1,...,K}$
8. Resultado $C = \{C_j\}_{j=1,...,K}$

4. EXPERIMENTOS COMPUTACIONALES

Para validar nuestro método, realizamos una serie de experimentos computacionales con datos usados en [3, 14, 22, 15] y comparamos los resultados con otros métodos de la literatura.

4.1. Colecciones de datos

Los datos que usamos para la evaluación de nuestro método son: Iris y Glass [3]; Crude Oil [14], Vowel [22] y Ruspini [15]. A continuación presentamos una breve descripción de las colecciones de datos usadas:

Iris. Compuesta por 150 muestras plantas de lirio. El número de atributos es cuatro. El valor de K para este conjunto de datos es 3.

Glass. Conformado por 214 instancias de diferentes tipos de vidrio. El número de atributos es nueve. El valor de K para este conjunto de datos es 7.

Crude Oil. Compuesta por 56 muestras de petróleo crudo. Los datos presentan seis atributos correspondientes al tipo de roca. El valor de K para este conjunto de datos es 3.

Vowel. Formada por 871 muestras de seis clases de vocales del dialecto hindú telugu. Los atributos corresponden a la medición de tres frecuencias. El valor de K para este conjunto de datos es 6.

Ruspini. Compuesta por 75 vectores en dos dimensiones formando cuatro grupos de manera natural. El valor de K para este conjunto de datos es 4.

4.2. Calibración del parámetro α

Los experimentos fueron realizados para cada uno de los conjuntos de datos considerando cuatro distintos valores para α : 0.25, 0.50, 0.75 y 1.00. Con cada valor de k se realizaron 50 ejecuciones del proceso GraspKM, con un valor de MAX_ITER de 100. Se debe tener en cuenta que, por las restricciones impuestas en la fase ConstruccionKM para la conformación del conjunto RCL, cuando α = 1, no significará que la solución sea construida de manera aleatoria, sino que el RCL estará conformado por todos los clusters cuyos centros sean más cercanos a un objeto x que el centro de su cluster.

En el Cuadro 1 se resumen los resultados obtenidos para los conjuntos de datos descritos. El cuadro muestra, el mejor valor, el peor valor y el promedio obtenido para la función objetivo f, en 50 ejecuciones de GraspKM.

Nótese, que el mejor valor es alcanzado cuando α = 1, por lo tanto es el valor que usaría en las comparativas con otras meta heurísticas.

4.3. Comparación de resultados

Las comparativas serán con respecto al mejor valor encontrado para la minimización de la función objetivo (1) frente a otras dos meta heurísticas de la literatura de clustering: el algoritmo genético KGA-Clustering propuesto por Bandyopadhyay y Maulik [1] y el algoritmo Grasp Grasp-KMeans propuesto por Cano *et al.* [6].

Sabemos que cada método presenta sus características propias, el hecho de establecer el número de ejecuciones y/o el máximo de iteraciones similares es con la intención de establecer condiciones parecidas para la ejecución de los métodos; en todo caso, la comparación se realizará principalmente en los valores obtenidos para la función objetivo.

Bandyopadhyay y Maulik presentan los resultados obtenidos para 50 ejecuciones de KGA-Clustering con un máximo de 1000 generaciones y población de 50 individuos para las colecciones de datos: Iris, Vowel y Crude Oil. Para las comparativas con el método GraspKM también consideraremos 50 ejecuciones con valor de α = 1 y MAX_ITER de 1000. Los resultados son resumidos en el Cuadro 2.

Cuadro 1. Calibración de α para la colección de datos Iris, Glass, Crude, Vowel y Ruspini.

	α	0.25	0.50	0.75	1.00
Iris	Mejor Óptimo	97.32594	97.32594	97.32594	**97. 22213**
	Peor Óptimo	97.32594	97.32594	97.32594	97.257195
	Promedio	97.325966	97.325966	97.325966	97.226074
Glass	Mejor Óptimo	201.12637	201.12637	201.12637	**200.58238**
	Peor Óptimo	201.83528	201.83528	201.85007	201.85007
	Promedio	201.15884	201.17648	201.16087	201.02412
Crude Oil	Mejor Óptimo	279.27097	279.27097	279.27097	**278.96515**
	Peor Óptimo	279.27097	279.27097	279.27097	278.96515
	Promedio	279.27097	279.27097	279.27097	278.96515
Vowel	Mejor Óptimo	149374.36	149374.36	149374.36	**149350.52**
	Peor Óptimo	149388.86	149398.67	149388.86	149400.92
	Promedio	149380.64	149380.64	149381.42	149379.84
Ruspini	Mejor Óptimo	504.77338	504.77338	504.77338	**501.27274**
	Peor Óptimo	504.77338	504.77338	504.77338	501.27274
	Promedio	504.77338	504.77338	504.77338	501.27274

En la tabla también se presentan los resultados para 50 ejecuciones del algoritmo K-Means, siendo superado por GraspKM en las tres colecciones de datos en cuanto al valor de la función objetivo y a la obtención de resultados más estables (véase promedio para K-Means y GraspKM). En comparación con KGA-Clustering, el método GraspKM se aproxima a KGA-Clustering en la colección de datos Iris. Precisamos decir que para la colección Crude Oil, tanto KGA-Clustering como GraspKM han obtenido los mismos resultados, y para la colección Vowel, GraspKM superó ampliamente los resultados de KGA-Clustering. Nótese que inclusive con un valor de MAX_ITER = 100 (véase el Cuadro 1) se han tenido resultados comparables con el método KGA-Clustering.

Cano *et al.,* presentan los resultados obtenidos para 10 ejecuciones del método Grasp-KMeans (con un máximo de 16 iteraciones) para las colecciones Glass y Ruspini. Para las comparativas con el método GraspKM también consideraremos 10 ejecuciones con MAX_ITER = 16. Los resultados son resumidos en el siguiente Cuadro 3.

El valor del Peor Óptimo no es mostrado por Cano *et al.,* por tanto no hay punto de comparación sobre este valor. En tanto que para los valores de Mejor Óptimo y Promedio el método GraspKM demuestra ser superior a Grasp-KMeans en las colecciones Glass y Ruspini.

5. CONCLUSIONES Y TRABAJOS FUTUROS

El presente artículo propone una metaheurística GRASP denominada GraspKM para el problema del clustering. El método está basado en los principios del algoritmo K-Means y ha demostrado ser superior al algoritmo K-Means y otras meta heurísticas. El método GraspKM se concentra en dos de los principales inconvenientes del algoritmo K-Means. El inconveniente de la convergencia a óptimos locales es acometida con la forma golosa adaptativa de construir soluciones. Esta característica, a su vez, alivia el inconveniente de la inicialización, al generar soluciones iniciales de buena calidad, que ayuda a la fase MejoriaKM a encontrar una mejor solución.

En la fase MejoriaKM se introduce el concepto de dispersión de un cluster, de manera que la reasignación de sus objetos bajo un criterio heurístico favorece a mejorar el valor de la función objetivo f.

La meta heurística Grasp es una secuencia repetitiva de procesos, de manera que a mayor cantidad de las ejecuciones existe la posibilidad de encontrar mejores soluciones. El clustering por lo general debe manejar grandes cantidades de objetos. En ese sentido, pueden usarse estructuras de datos que agilicen los accesos a los objetos (por ejemplo KD-Trees). También puede ser aplicada una reducción del número de cálculos para mejorar los tiempos de respuesta cuando la cantidad de objetos es grande (derivados del teorema de la desigualdad de triángulos).

AGRADECIMIENTOS

Al Instituto de Investigación de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos de Lima-Perú, por el apoyo brindado para la publicación de esta investigación.

Cuadro 2. Comparativa entre KGA-Clustering y GraspKM para la colección de datos Iris.

	α	K-Means	KGA-Clustering	GraspKM
Iris	Mejor Optimo	97.32594	97.100777	97.22213
	Peor Optimo	128.40422	97.100777	97.22213
	Promedio	104.26579	97.100777	97.22213
Crude Oil	Mejor Optimo	279.270997	278.965150	278.965150
	Peor Optimo	359.761992	278.965150	278.965150
	Promedio	284.248060	278.965150	278.965150
Vowel	Mejor Optimo	149399.49	149356.01	149342.64
	Peor Optimo	168764.32	149378.03	149374.36
	Promedio	154150.77	149368.45	149360.25

Cuadro 3. Comparativa entre Grasp-KMeans y GraspKM para la colección de datos Glass y Ruspini.

	α	K-Means	KGA-Clustering	GraspKM
Glass	Mejor Optimo	203.865	204.992	201.126
	Promedio	212.591	205.239	202.233
Ruspini	Mejor Optimo	864.223	720.142	501.272
	Promedio	1178.064	720.142	502.028

REFERENCIAS BIBLIOGRÁFICAS

[1] Bandyopadhyay, S., Maulik, U. «An evolutionary technique based on K-Means algorithm for optimal clustering in Rn». *Information Sciences.* Vol. 146 (1-4), 2002, pp. 221–237.

[2] Bathia, S., Deogun, J. Conceptual Clustering in Information Retrieval. IEEE Transactions on Systems, Man, and Cybernetics, Part B: *Cybernetics.* Vol 28 (3), 1998, pp. 427–436.

[3] Blake, C., Merz, C. UCI Repository of machine learning databases [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: Uinversity of California, Department of Information and Computer Science, 1998.

[4] Bezdek, J. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[5] Brucker, P. «On the Complexity of Clustering Problems». *Lecture Notes in Economics and Mathematical Systems.* Vol. 157, 1978, pp. 45–54.

[6] Cano J., Cordon, O., Herrera, S. Sanchez L. Greedy Randomized Adaptive Search Procedure Applied to the Clustering Problem as an Initialization process Using K-Means as a Local Search Procedure *International Journal of Intelligent and fuzzy Systems*. Vol. 12, 2002, pp. 235-242.

[7] Dempster, A. Laird, N. Rubin, D. Maxinum-likelihood from incomplete data via the em *algorithm. Journal of the* Royal Statiscal Society B. Vol. 39, 1977, pp. 1-39.

[8] Fayyad, U. Piatetsky-Shapiro, G. Padhraic S. From data Mining to Knowledge Discovery in Databases. *American Association for Artificial Inteligence* 1996, pp. 37-54.

[9] Feo, T., Resende, M. Greedy Randomized Adaptative Search Procedure. Journal of Global Optimization Vol. 6, 1995, pp. 109-133.

[10] Forgy, E. «Cluster analysis of multivariate data: E_ciency vs. Interpretability of classifications». *Biometrics.* Vol. 21 (768), 1965.

[11] Fränti, P., Kivijärvi, J. «Randomised Local Search Algorithm for the Clustering Problem». Springer-Verlag London Limited. *Pattern Analysis and Applications.* Vol 3, 2000, pp. 358–369.

[12] Hansen P., Mladenovic N. J-Means: A new local search heuristic for minimun sum-of-squares clustering. Pattern Recognition. Vol. 34 (2), 2001, pp. 405–413.

[14] Johnson, R., Wichern, D. Applied Multivariate Statistical Analysis. Prentice Hall, Englewood Cli_s, NJ. 1982. [15] Kaufman, L., Rousseeuw, P. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons. New York, NY. 1965.

[15] Kaufman, L., Rousseeuw, P. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons. New York, NY. 1965.

[16] Kearns, M., Mansour, Y., Ng, A. An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering. Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann. 1997, pp. 282–293.

[17] Makhoul, J., Roucos, S., Gish, H. Vector quantization in speech coding. Preccedings of the IEEE. Vol. 73, 1985, pp. 1551–1558.

[18] McQueen, J. Some methods for classification and analysis of multivariate observations. In Preccedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. 1967, 281–297.

[19] Merz, P. Analysis of gene expression profiles: an application of memetic algorithms to the minimum sum-of-squares clustering problem. BioSystems. Vol. 72 (1-2), 2003, pp. 99–109.

[20] Murthy, C. Chowdhury, N. In search of optimal clusters using genetic algorithms. *Pattern Recognition Letters.* Vol. 17, 1996, pp. 825-832.

[21] Pacheco, J. Valencia, O. Design of hybrids for the minimin sum-of-squares clustering problem. *Computational Statistics Data Analysis.* Vol. 43 (2), 2003, pp. 235-248.

[22] Pal, S., Dutta, D. Fuzzy sets and decision making, approaches in vowel and speaker recognition. *IEEE transaction on Systems, Man, and Cybernetics*. Vol. 7, 1977, pp 625-629.

[23] Peña, J., Lozano J., Larrañaga, P. An empirical comparison of four initialization methods for the KMeans algorithm. Pattern Recognitioin Letters. Vol. 20, 1999, pp. 1027–1040.

[24] Pham, D., Prince, J. An Adaptive Fuzzy C-Means Algorithm for Image Segmentation in the Presence of Intensity Inhomogeneities. Pattern Recognition Letters. Vol. 20 (1), 1999, pp. 57–68.

[25] Scheunders, P. A genetic Lloyd-Max image quantization algorithm. Pattern Recognition Letters. Vol. 17 (5), 1996, pp. 547–556.

[26] Solberg, A., Taxt, T., Jain, A. A markov random field model for classification of multisource satellite imagery. IEEE Trans. Geoscience and Remote Sensing. Vol. 34 (1), 1996, pp. 100–113.