ALGORITMOS METAHEURISTICOS BASADOS EN LA LEY DE GRAVITACION UNIVERSAL PARA AGRUPACION DE DATOS

Angélica J. SUAREZ
Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas
Bogotá, Cundinamarca, Colombia

y

Jorge E. RODRIGUEZ
Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas
Bogotá, Cundinamarca, Colombia

RESUMEN

El presente documento contiene una revisión del estado del arte de la aplicación de técnicas metaheurísticas basadas en la ley de gravitación universal a la tarea de agrupación de minería de datos, la motivación fue originada por la identificación de las dos tendencias por separado (algoritmos metaheurísticos y basados en la ley de gravitación universal) en las recientes investigaciones relacionadas con agrupación de datos y los resultados sobresalientes que estas propuestas han logrado en comparación a las técnicas de minería de datos convencionales.

Palabras Claves: Minería de Datos, Agrupación de Datos, Ley de Gravitación Universal, Algoritmos Metaheurísticos.

1. INTRODUCCIÓN

El termino metaheurístico fue introducido en 1986 por Golver en su estudio sobre los que en su momento consideró como desarrollos innovadores con gran potencial para resolver problemas de optimización desde las perspectivas de la investigación de la inteligencia artificial [1]. En los años posteriores este tipo de algoritmos han ganado gran popularidad y demostrando su capacidad de resolver una amplia gama de problemas en variados dominios de aplicación, siempre caracterizándose por su capacidad de hacer frente a problemas complejos con poco o ningún conocimiento del espacio de búsqueda, característica relevante en problemas de optimización [2]. En paralelo propuestas inspiradas en la ley de gravitación universal han resuelto de manera sobresaliente el problema de la agrupación de datos, el objetivo de investigación del presente documento fue determinar por una revisión de la literatura si el desarrollo de técnicas metaheurísticas basadas en la ley de gravitación universal ha aportado a la solución del problema de agrupación de datos y pueden ser consideradas como iniciativas con gran potencial para la solución de este tipo de problema y así despertar el interés de la comunidad científica hacia estas.

2. DESCRIPCION DEL PROBLEMA

La minería de datos se ha convertido en la herramienta predilecta para la extracción de conocimiento en las condiciones que la actual sociedad de la información requiere, gracias a su fuerte capacidad de procesar de manera automática o semiautomática grandes cantidades de datos almacenados en distintos formatos para extraer conocimiento nuevo y útil oculto en los datos [3], cada vez es más amplio el dominio de aplicación en el que han sido utilizadas sus técnicas con resultados satisfactorios. Por otro lado la agrupación es una de las tareas de minería de datos más importantes, se refiere al proceso de categorización de los objetos de datos en grupos con objetos similares entre ellos y diferentes a los objetos pertenecientes a otros grupos, en la actualidad existe una amplia variedad de métodos de agrupación con diferentes enfoques, sin embargo es bien conocido que estos métodos tienen diferentes limitaciones. En recientes investigaciones algunas de estas limitaciones fueron abordadas por medio del uso de técnicas de optimización metaheurísticas, propuestas que han demostrado que aportan mejoras en las tareas de agrupación en cuanto a calidad, complejidad, tiempo y costo computacional.

Adicionalmente las recientes propuestas de técnicas de agrupación basadas en la ley de gravitación universal [4], [6], [9], [10], [11], [12], [13] también han demostrado resolver el problema de la agrupación de datos de manera eficaz entregando mejores resultados que las técnicas convencionales de minería de datos, el objetivo de esta investigación es realizar una revisión de estado del arte e identificar como la integración de las técnicas metaheurísticas y basadas en la gravitación han aportado para la solución del problema de la agrupación de datos con el propósito de servir como fundamento para futuras investigaciones que se relacionen con los temas considerados.

3. ALGORITMOS METAHEURISTICOS

Metaheurístico deriva del verbo griego “heuristikein” que significa “encontrar” y el prefijo “meta” que significa “en alto nivel”. En términos generales un algoritmo metaheurístico puede ser visto como una estrategia de alto nivel que desarrolla búsquedas aleatorias dirigidas dentro posibles soluciones para hallar la mejor solución (cercana a la óptima) de un problema, originalmente fueron definidos como métodos que orquestan una interacción entre procedimientos de mejora locales y estrategias de nivel superior para crear un proceso capaz de escapar de los óptimos locales y realizar una búsqueda robusta de un espacio de soluciones, posteriormente se incluyeron procedimientos que emplean estrategias para superar la trampa de los óptimos locales en espacios complejos de soluciones [1]. Los algoritmos metaheurísticos se caracterizan principalmente por [2]:

1) Son estrategias que "guían" el proceso de búsqueda.
2) Los conceptos básicos de un algoritmo metaheurístico se puede describir en un nivel abstracto no atado a un problema específico.

3) Los algoritmos más avanzados utilizan la experiencia de búsqueda (simulan algún tipo de memoria) para guiar la búsqueda.
4) Incorporan mecanismos para evitar caer en óptimos locales.

Los algoritmos metaheurísticos puede clasificarse considerando diferentes aspectos, por ejemplo: basados o no basados en la naturaleza, con memoria o sin memoria, con función objetiva estática o dinámica, sin embargo la clasificación más apropiada es aquella que considera la manipulación en cada iteración de un solo punto del espacio de búsqueda "*trayectoria*" o de un conjunto "*población*", El término "*trayectoria*" es utilizado ya que la búsqueda genera una trayectoria en el espacio de búsqueda, en otras palabras la búsqueda parte de un punto y mediante la exploración del vecindario va variando la solución actual formando así una trayectoria, generalmente surgen a partir de la mejora de métodos de búsqueda local al incorporarse técnicas que les permitan escapar de óptimos locales, los algoritmo basados en "trayectoria" incorporan criterios de terminación como un número máximo de iteraciones, identificación de un estancamiento o hallarse una solución lo suficiente aceptable.

Los algoritmos basados en "*población*" trabajan paralelamente con un conjunto de agentes (soluciones) en cada iteración permitiendo así una manera natural e intrínseca de explorar el espacio de búsqueda [2], por ejemplo, los algoritmos inspirados en el comportamiento de enjambres utilizan una colección de agentes (soluciones) similar a una bandada natural de aves o peces, donde cada miembro ejecuta una serie de operaciones particulares y comparte su información con los otros, estas operaciones son generalmente simples, sin embargo, su efecto colectivo, conocido como inteligencia de enjambre, produce un resultado sorprendente. Las interacciones locales entre agentes proporcionan un resultado global que permiten al sistema resolver el problema sin utilizar ningún controlador central. En este caso, las operaciones de los miembros, incluyendo la búsqueda al azar, la retroalimentación positiva, la retroalimentación negativa y múltiples interacciones, conducen a una situación de auto-organización [4].

Se puede reconocer dos tareas comunes en los algoritmos metaheurísticos basados en la población: la *exploración* y *explotación*. La *exploración* es la capacidad de sondear el espacio de búsqueda y la *explotación* es la capacidad de encontrar el óptimo alrededor de una buena solución. En las primeras iteraciones un algoritmo de búsqueda metaheurístico *explora* el espacio de búsqueda para encontrar nuevas soluciones esto le evita caer en un óptimo local, razón de la importancia de esta tarea, con el paso de las iteraciones, la *exploración* se desvanece y se pasa a la explotación, así el algoritmo va afinándose en puntos semi-óptimos. La clave esencial para tener una búsqueda de alto rendimiento es un adecuado equilibrio entre *exploración* y *explotación*, por un lado para identificar rápidamente regiones en el espacio de búsqueda con soluciones de alta calidad y por el otro para no perder demasiado tiempo en las regiones del espacio de búsqueda que ya se exploró o que no proporcionan soluciones de alta calidad. Todos los algoritmos metaheurísticos basados en la población emplean la *exploración* y la *explotación* pero utilizando diferentes enfoques y operadores.

Por otro lado los agentes de un algoritmo de búsqueda basado en población, pasan por tres pasos en cada iteración para realizar la exploración y explotación: auto-adaptación, cooperación y competición, en el paso de auto-adaptación cada miembro (agente) mejora su desempeño, en el paso de cooperación, los miembros colaboran con cada otro por transferencia de información y finalmente en el paso de competición, los miembros compiten por supervivir. Lo anterior nos lleva a concluir que todos los algoritmos metaheurísticos de búsqueda tienen un marco común.

Al revisar la literatura puede evidenciarse que por lo general los algoritmos metaheurísticos son inspirados en la naturaleza e imitan procesos físicos o biológicos, entre los más populares encontramos: el algoritmo de Optimización por Enjambre de Partículas *(Particle Swarm Optimization - PSO),* que simula el comportamiento de una bandada de aves; algoritmo Genético *(Genetic Algorithm - GA)*, inspirado en la teoría de la evolución de Darwin; el algoritmo de Simulación de Cocción *(Simulated Annealing - SA)*, inspirado en los efectos de la termodinámica; algoritmo de colonia de hormigas *(Ant Colony Optimization - ACO)*, que simula el comportamiento de una colonia de hormigas en busca de comida [3].

4. OPTIMIZACION EN AGRUPACION DE DATOS

La agrupación es una tarea fundamental del aprendizaje computacional y la minería de datos que toma un conjunto de datos y los clasifica en diferentes grupos en base a la similitud calculada entre ellos, de tal manera que en un mismo grupo se encuentran los objetos (registros de datos) más similares entre sí y diferentes a los objetos pertenecientes a otros grupos. Existe un gran número de algoritmos de agrupación, generalmente enfocados en métodos de partición o jerárquicos, ambos enfoques tienen sus propias ventajas y limitaciones en cuanto al número, forma y superposición de los grupos, En la actualidad algunas de las nuevas propuestas se enfocan en el uso de diferentes técnicas de optimización, la participación de técnicas de optimización inteligentes en las tareas de agrupación ha logrado encontrar formas eficaces de mejorar la complejidad, tiempo y costo de los procesos de minería de datos [4].

Matemáticamente, un problema de agrupamiento se puede definir como sigue, dado $O = \{O_1,\ldots,O_n\}$ donde O es un conjunto finito de n objetos (vector) en un espacio de elementos S, el objetivo de un problema de agrupación de datos es hallar la partición optima de los objetos $C = \{C_1,\ldots C_D\}$, $O = \cup_{i=1}^{p} C_i$, y $C_i \cap C_j = \varphi; para\ i \neq j$, Donde Ci representa el *i-ésimo* grupo de la partición C, de tal manera que los datos que pertenecen al mismo grupo son similares mientras que son lo más diferentes posible a los datos que pertenecen a otros grupos en términos de una función de medición de distancia. Las particiones resultantes son llamadas grupos y deben cumplir con las siguientes condiciones: a) cada grupo debe contener por lo menos un objeto; b) los diferentes grupos no deben tener objetos en común; c) Cada objeto debe ser asignado a un único grupo, en otras palabras, después de asignar objetos a los grupos, la suma de los objetos de todos los grupos debe ser igual al número de objetos del conjunto de datos original [5]. Sin embargo es posible generar diferentes particiones no óptimas del conjunto de datos que satisfagan las condiciones antes mencionadas, se hace necesario entonces optimizar la calidad de la partición obtenida.

Como es sabido el objetivo básico de un algoritmo de optimización es minimizar o maximizar una función objetivo eligiendo sistemáticamente valores tomados desde un espacio de búsqueda, para el problema de agrupación de datos se debe seleccionar una función objetivo que permita evaluar la calidad

de la partición obtenida. La función más popular para esto es el error medio cuadrático que considera la cohesión de los grupos en orden a evaluar la calidad de una partición dada:

$$f(O,C) = \sum_{i=1}^{D} \sum_{O_j \in C_i} \|O_j - Z_i\|^2 \quad (1)$$

Donde D es el número de grupos y $\|O_j - Z_i\|^2$ es la *distancia euclidiana* entre un objeto de datos $O_j \in C_i$ y el centro del grupo i, representado por el símbolo Z_i, que puede ser calculado a partir de la siguiente ecuación:

$$Z_i = \frac{1}{|C_i|} \sum_{k \in C_i} O_k \quad (2)$$

Donde $|C_i|$ es la cardinalidad del grupo C_i, es decir el número de objetos que posee el grupo *i.*

En los problemas de agrupación el objetivo puede ser hallar el centroide de cada grupo por medio de la *minimización* de una función objetivo como la suma de las distancias entre cada objeto y el centro del grupo al que está asignado expresada por el *error medio cuadrático* calculado por la Ec. (1).

5. LEY DE GRAVITACION UNIVERSAL

La ley de gravitación universal fue publicada en 1687 por Isaac Newton en su libro "*Philosophiae Naturalis Principia Mathematica"*. Esta ley enuncia una relación cuantitativa de la interacción gravitatoria entre distintos cuerpos con masa, define que la fuerza con que se atraen dos cuerpos de diferente masa únicamente depende del valor de sus masas y del cuadrado de la distancia que las separa y que dicha fuerza actúa de tal forma que es como si toda la masa de cada uno de los cuerpos estuviese concentrada únicamente en su centro. Es decir, cuanto mayor masa tengan los cuerpos y más cercanos se encuentren, con mayor fuerza se atraerán [6].

La fuerza de Gravitación Universal se calcula como:

$$F(t) = \frac{G m_x m_y}{d(x(t), y(t))^2} \quad (3)$$

Dónde,
m_x: Es la masa del primer cuerpo
m_y: Es la masa del segundo objeto
d: es la distancia entre los cuerpos
$F(t)$: es el módulo de la fuerza ejercida entre ambos cuerpos, y su dirección se encuentra en el eje que une ambos cuerpos.
G: Es la constante de la Gravitación Universal, igual a:

$$\boldsymbol{G = (6.67428 \pm 0.00067) \times 10^{-11} Nm^2 kg^{-2}} \quad (4)$$

Un algoritmo metaheurístico basado en la ley de gravitación universal considera que los objetos (soluciones) se atraen entre sí por la fuerza de la gravedad y esta fuerza provoca un movimiento global de todos los objetos hacia los objetos con masas más pesadas, por lo tanto las masas cooperan usando una comunicación directa a través de la fuerza de gravedad, las masas más pesadas que corresponden a las mejores soluciones, se mueven más lentamente que las masas ligeras, esto garantiza la etapa de *explotación* en el algoritmo.

6. ALGORIRMOS DE AGRUPACION GRAVITACIONALES

En 1977 W.E. Wright inicio la línea de los modelos de agrupación basados en la ley de gravitación universal, presentando un algoritmo de agrupamiento denominado "Gravitational clustering", diseñó un algoritmo para desarrollar análisis de agrupamiento en datos euclidianos, evalúo los resultados mediante su aplicación en varios conjuntos de datos y los comparó con los obtenidos a través de algoritmos no gravitacionales con resultados exitosos [7], el algoritmo gravitacional propuesto por Wright es un algoritmo jerárquico de aglomeración, las fuerzas gravitacionales son usadas como mecanismos para unir partículas hasta que una sola partícula continua en el sistema [7].

En 1999 S. Kundu, desarrolló un método de agrupación basado en la noción de la existencia de una fuerza de atracción gravitacional entre cada par de puntos, con la novedad de no utilizar una medida de "*similitud*", Los grupos se forman al permitir que cada punto se mueva lentamente bajo el efecto resultante de todas las fuerzas que actúan sobre él y mediante la fusión de dos puntos cuando están muy cerca uno del otro. Este modelo se consideró como un refinamiento del método del vecino más cercano y el método difuso de K-medias [8].

En 2003, J. Gómez, D. Dasgupta y O. Nasraoui [6], propusieron un nuevo algoritmo de agrupación gravitacional no-supervisado que se nombra para efectos de esta investigación como "NGCA", este método determina automáticamente el número de clases, cada objeto en la base de datos es considerado como un objeto en el espacio y los objetos se mueven utilizando la ley de gravitación universal y la segunda ley de Newton, el algoritmo propuesto funciona correctamente en datos con ruido y está fundamentada en la investigación desarrollada por Wright, mejorando su rapidez, robustez además de logar un algoritmo no-supervisado.

En 2012, M. Sánchez y O. Castillo [9], desarrollaron un algoritmo para la búsqueda de grupos, también basado en la ley de la gravitación universal de Newton "FGGC", este incluyo la teoría difusa para refinar los grupos de salida. En términos generales incorpora al análisis de agrupamiento la granularidad difusa que tiene como principio obtener el gránulo óptimo que representa completamente el conocimiento del conjunto de datos. La computación granular ha ido ganando mucho interés en los últimos años y este trabajo continúa esta tendencia en el área.

En 2014, A. Hashempour y H. Nezamabadi-pour, propusieron un nuevo algoritmo denominado "*Gravitational Ensemble Clustering - GEC*" [10], El método propuesto combina los resultados obtenidos por diferentes algoritmos de agrupamiento como el algoritmo K-medias utilizando conceptos gravitacionales, con el objetivo de superar las debilidades de los algoritmos individuales y mejorar su desempeño. Los resultados experimentales evidenciaron su versatilidad, robustez y capacidad de generar mejores resultados que los algoritmos utilizados por separado.

7. ALGORIRMOS METAURISTICOS GRAVITACIONALES PARA AGRUPACION

En la actualidad son escasas las propuestas de algoritmos metaheurísticos basados en la gravitación universal, En 2009 E .

Rashedi, H. Nezamabadi-pour y S. Saryzdi propusieron el Algoritmo de Búsqueda Gravitacional (*Gravitational Search Algorithm- GSA)* el único encontrado en la revisión de literatura con estas características. En el algoritmo GSA [4], cada masa (agente) tiene cuatro especificaciones: posición, masa de inercia, masa gravitacional activa y masa gravitacional pasiva. La posición de la masa corresponde a una solución del problema y las masas de inercia y gravitacional son determinadas utilizando una función objetivo (función fitness), en otras palabras, cada masa representa una solución y el algoritmo navega ajustando correctamente las masas gravitacionales y de inercia, con el transcurso del tiempo, se espera que las masas sean atraídas por las más pesadas, estas masas representan una solución óptima en el espacio de búsqueda.

Considerando, un sistema con *N* agentes (masas). Se define la posición del agente *i-ésimo* por:

$$X_i = \left(x_i^1, \dots, x_i^d, \dots, x_i^n\right) para\ i = 1,2, \dots, N \tag{5}$$

Donde x_i^d presenta la posición del agente *i–esimo* en la dimensión *d-esima*. En un momento determinado *'t'*, la fuerza que actúa sobre la masa *'i'* a partir de la masa *'j'* es la siguiente:

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}(x_j^d(t) - x_i^d(t)) \tag{6}$$

Donde M_{aj} es la masa gravitacional activa relacionada con agente *j*, M_{pi} es la masa gravitatoria pasiva relacionada con el agente *i*, *G(t)* es constante gravitacional en el tiempo *t, ε* es una constante pequeña y R_{ij} (t) es la distancia euclidiana entre los agentes *i* y *j*:

$$R_{ij}(t) = \left\|X_i(t), X_j(t)\right\|_2 \tag{7}$$

Para dar características estocásticas al algoritmo, se supone que la fuerza total que actúa sobre el agente *i* en una dimensión *d* corresponde a una suma ponderada al azar de los componentes de los *d-ésimos* componentes de las fuerzas ejercidas por los otros agentes:

$$F_i^d(t) = \sum_{j=1 j \neq i}^{N} rand_j F_{ij}^d(t) \tag{8}$$

Donde $rand_j$ es un número aleatorio entre el intervalo [0, 1]. Por lo tanto, por la ley del movimiento, la aceleración del agente *i* en el momento *t*, y en la dirección *d-ésima*, $a_i^d(t)$ corresponde a:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{9}$$

Donde M_{ii} es la masa inercial del agente *i.* Además, la siguiente velocidad de un agente se considera como una fracción de su velocidad actual más su aceleración. Por lo tanto, su posición y su velocidad corresponde a:

$$\begin{aligned} v_i^d(t+1) &= rand_i \times v_i^d(t) + a_i^d(t) \\ x_i^d(t+1) &= x_i^d(t) + v_i^d(t+1) \end{aligned} \tag{10}$$

Donde $rand_i$ es una variable aleatoria entre el intervalo [0, 1]. Utilizando este número aleatorio se agrega una característica al azar a la búsqueda [4].

Las masas gravitatorias y de inercia son simplemente calculadas por la evaluación de la función fitness (función objetivo – minimización\maximización). Una masa más pesada significa un agente más eficiente.

$$M_{ai} = M_{pi} = M_{ii} = M_i\ , \qquad i = 1,2, \dots \dots N$$

$$\begin{aligned} M_i(t) &= \frac{fit_i(t) - worst\ (t)}{best(t) - worst\ (t)} \\ M_i\ (t) &= \frac{m_i(t)}{\sum_{j=1}^{N} m_i(t)} \end{aligned} \tag{11}$$

Donde $fit_i\ (t)$ representa el valor de la adecuación del agente *i* en el momento *t*, y *worst(t)* y *best(t)* se definen de la siguiente manera (para problemas de minimización):

$$\begin{aligned} best(t) &= \min_{j \in \{1, \dots N\}} Fit_i(t) \\ worst(t) &= \max_{j \in \{1, \dots N\}} Fit_j(t) \end{aligned} \tag{12}$$

Es importante resaltar que con el fin de evitar la captura en un **óptimo local** el algoritmo utiliza el principio de *exploración*. En el transcurso de las iteraciones, la *exploración* debe desaparecer y *explotación* debe aparecer gradualmente. Para mejorar el rendimiento de GSA controlando la *exploración* y *explotación* sólo los agentes *Kbest* atraerán a los demás. Por lo tanto, la Ec. (8) podría ser modificado como:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_i F_{ii}^d(t) \tag{13}$$

Por otra parte se identificaron dos adaptaciones del algoritmo GSA para la agrupación de datos, los algoritmos GSA-HS [12] y GGSA [13].

En 2011, A. Hatamlou, S.Abdullah y Z. Othman, propusieron el *Algoritmo de Búsqueda Gravitacional con Heurísticas para la agrupación de datos GSA- HS*, en el cual el algoritmo GSA es utilizado para encontrar la solución más cercana a la óptima y posteriormente se aplica un algoritmo de búsqueda heurística para mejorar la solución inicial mediante la búsqueda en torno a ella [12].

El algoritmo de búsqueda heurística utilizado funciona en términos generales validando el efecto que tiene la adición de una constante de movimiento a cada atributo de cada centroide y recalculando valor de la función objetivo (función fitness) para el nuevo centroide generado por la adición, si hay mejora el centroide es reemplazado por el nuevo centroide, de lo contrario se cambia el sentido de búsqueda, lo que significa que el valor de la constante ahora debe ser restado al valor del atributo y realizar la misma validación, si no hay mejoría en ambos lados del atributo actual en el centroide actual para el valor de la constante actual, el valor de la constante se divide en dos para la siguiente iteración. El anterior proceso se repite para todos los atributos del centroide y luego para otros centroides secuencialmente hasta que los criterios de terminación son alcanzados.

Tabla 1. Pseudocódigo Algoritmo GSA-HS

Paso 1: Método GSA
1.1. Generar la población inicial usando los datos de prueba
1.2. Evaluar el valor de la función fitness para la población
1.3. Calcular M, F, a para la población, Ec. (6, 8 y 9) del algoritmo GSA
1.4. Actualizar la velocidad y posición de la población, Ec. (10) del algoritmo GSA
1.5. Si el criterio de terminación es alcanzado pasar el Paso 2
Paso 2: Búsqueda Heurística
For all *centroides i=1...k do*
  For all *Atributos j=1...d do*
  If *SDi (j) == 1*
  $C_i(j) = C_i(j) + SSi(j)$;
  Calcular el valor de la función fitness para el nuevo centroide
  If *el valor de la función fitness presenta mejora*
  Hacer el Nuevo centroide permanente
  Else
  Recargar el centroide previo
  $SD_i(j) == -1$
  End if

```
Else
 if SDx (j) == -1
  Calcular el valor de la función fitness para el Nuevo centroide
  If el valor de la función fitness presenta mejora
  Hacer el Nuevo centroide permanente
  Else
   Recargar el centroide previo
   SDi (j) == 0
  End if
 Else if SDi (j) == 0
  SSi(j) = SSi(j)/2;
  SDi (j) == 1;
 End if
End for
End for
```

En la Tabla 1. Se encuentra el pseudocódigo del Algoritmo GSA-HS, Donde, $SD= [SD_1, SD_2,..., SD_k]$ es la dirección de búsqueda, $SS= [SS_1, SS_2,..., SS_k]$ es el paso de la búsqueda, $SD_i= [1,1,..,1]$ es la dirección de búsqueda para el *i-ésimo* centroide y *d* corresponde a la longitud de este arreglo que es igual a la dimensión del conjunto de datos, SS_i es el paso de búsqueda para el *i-ésimo* centroide. Los resultados logrados por el algoritmo fueron de alta calidad superando a los resultados generados por los algoritmos k-medias y PSO (*Particle Swarm Optimization*).

En 2014, M. Dowlatshahi y H. Nezamabadi-pour, propusieron el *Algoritmo de Búsqueda Gravitacional para la Agrupación de Datos (Grouping Gravitational Search Algorithm - GGSA)* [13], que es una adaptación para resolver el problema de agrupación de datos del Algoritmo de Búsqueda Gravitacional – GSA, , los cambios realizados para lograr el anterior objetivo fueron principalmente dos, el primero consistió en la integración de un esquema de codificación necesario para lograr al algoritmo trabajar con las posibles soluciones de un problema de agrupación y el segundo consistió en la modificación algunas de las ecuaciones del GSA para adoptarlo al mencionado esquema de codificación.

La representación usada por el algoritmo GGSA es una representación de agrupamien construida por dos partes: *parte ítem y parte grupo,* la *parte ítem* consiste en un arreglo de tamaño *n* (n es el número de objetos). La *parte grupo* consiste en una permutación de *D* etiquetas de grupos, Cada miembro en la *parte ítem* puede tener cualquiera de la etiquetas de grupo, indicando que el objeto pertenece al grupo indicado por la etiqueta, en la siguiente ilustración se muestra la codificación utilizada por el algoritmo GGSA para una solución $C = \{C_1\{ O_1, O_3\}, C_2\{ O_2, O_4, O_5\}\}$ de un problema de datos con $O= \{ O_1, O_2 O_3, O_4, O_5\}$.

Cuando se optimiza una función por el algoritmo GSA, por lo general cada solución es representada por un vector de longitud *D* de números reales (*D* es la dimensión del campo de búsqueda), donde cada valor corresponde a una variable, de manera similar cuando se resuelve un problema de agrupación con el algoritmo GGSA, se puede representar una solución compuesta de *D* grupos como una estructura cuyo largo es igual al número de grupos, en otras palabras los grupos juegan un rol de variables en el estándar GSA, de manera similar la posición de cada objeto en la *d-ésima* dimensión representa el valor en la *d-ésima* variable, en el algoritmo GGSA este determina los datos que están contenidos en el *d-ésimo* grupo [13].

Ilustración 1. Codificación de una solución candidata de dos grupos para un problema de agrupación de cinco objetos de datos

Parte Item

1	2	1	2	2
O_1	O_2	O_3	O_4	O_5

Parte grupo

1	2
C_1	C_2

La distancia entre grupos es calculada por medio del coeficiente de similitud de *Jaccard.* Dados dos grupos C_1 y C_2 con cardinalidad $|C_1|$ y $|C_2|$ respectivamente, el coeficiente se calcula a partir de la siguiente ecuación:

$$Dist_j(C_1, C_2) = Dist_j(C_2, C_1) = 1 - \frac{|C_1 \cap C_2|}{C_1 \cup C_2} \quad (14)$$

Donde $Dist_j(C_1, C_2)$ es el grado de disimilitud entre los dos grupos C_1 y C_2 y permite determinar cuán lejos están, en general se tiene que el $0 \leq Dist_j(C_1, C_2) \geq 1$, donde el valor será igual a 0 si $C_1 = C_2$ e igual a 1 si $C_1 \cap C_2 = \varphi$.

La distancia Euclidiana se adaptó también para trabajar con problemas de agrupación de datos, Dado $C=\{C_1, \dots, C_D\}$ y $C'=\{C'_1, \dots, C'_D\}$ son dos grupos (clusters) candidatos de objetos de datos, se tiene que la distancia entre C y C' es:

$$Euclidean_j(C, C') = Euclidean_j(C', C)$$
$$= \sqrt{Dist_j(C_1, C'_1)^2 + \cdots + Dist_j(C_D, C'_D)}$$
$$= \sqrt{\sum_{i=1}^{D} Dist_j(C_i, C'_i)^2} \quad (15)$$

Ya que cada operación se realiza entre un par de grupos (clusters), para que le resultado del cálculo sea apropiado cada par de grupos debe ser el par más similar, para esto se usa el pareo por peso máximo bipartido *(Maximum Weight bipartite Matching - MWM)* aplicado en [13].Que consiste en una metodología que permite organizar pares por similitud.

$$v_i^d(t+1) = rand \times v_i^d(t) + G(t) \sum_{j \in Kbest, j \neq i} rand \frac{M_j(t)}{Euclidean(X_i(t), X_j(t) + \varepsilon)} \times Dist_j(x_j^d(t), x_i^d(t)) \quad (16)$$

La construcción del nuevo cluster $x_i^d(t+1)$, durante la fase de herencia, debe ser de tal manera que el grado de disimilitud con $x_i^d(t))$ este cercano al valor de $v_i^d(t+1)$, lo que significa que el grado de disimilitud entre los dos grupos debe ser al menos igual a $1 - v_i^d(t+1)$, en otras palabras se busca el número de ítems compartidos entre $x_i^d(t+1)$ y $x_i^d(t)$ es igual a $n_i^d(t+1) = |x_i^d(t+1) \cap x_i^d(t)|$ del tal manera que el valor de $Dist_j(x_i^d(t+1), x_i^d(t))$ se aproxime al valor de $v_i^d(t+1)$, de hecho los números compartidos entre $x_i^d(t+1)$ y $x_i^d(t)$ son una de las partes de la solución $X_i(t+1)$ heredados desde $X_i(t)$, a partir de la Ec. (16) el número de ítems compartidos entre $x_i^d(t+1)$ y $x_i^d(t)$ es calculado de la siguiente manera:

$$Dist_j\left(x_i^d(t+1), x_i^d(t)\right) = \frac{n_i^d(t+1)}{|(x_i^d(t)|} \approx v_i^d(t+1)$$
$$\Rightarrow n_i^d(t+1) \approx (1 - v_i^d(t+1))|x_i^d(t)| \quad (17)$$

Tabla 2. Pseudocódigo Algoritmo GGSA

```
// Inicialización
Generar la Inicial población;
// Iteracion
For i = 1 to #_iteraciones
 Evaluar la función fitness para cada agente;
 Actualizar G, best y worst de la población;
 Calcular M;
// Ordenamiento pares - MWM
For d = 1 to D do
```

Par de cluster d of $X_i(t)$ con el cluster mas similar de $X_j(t)$
(para todo $X_j(t) \in$ Kbest) por el procedimiento de pareo MWM;

// **Fase de herencia**
Calcular $Euclidean(X_i(t), X_j(t))$ (para todo $X_j(t) \in$ Kbest) usando la Ec.(14);
For d = 1 to D do
Calcular $Dist_j(x_j^d(t), x_i^d(t))$ (para todo $X_j(t) \in$ Kbest) usando la Eq.(13);
Calcular el valor de $v_i^d(t+1)$ usando la Ec. (15);
Calcular el valor de $n_i^d(t+1)$ Ec. (16);
Seleccionar aleatoriamente $n_i^d(t+1)$ desde el cluster $X_i(t)$ y asignarlos al nuevo cluster $X_i(t+1)$;
Endfor
// **Fase de inserción**
For each objeto O_j que no fue seleccionado en la fase de herencia **do**
Asignar el objeto O_j en el cluster con el centroide más cercano;
Output: $X_i(t+1)$
// **End Iteracion**
Endfor

Los resultados experimentales confirmaron la eficacia del algoritmo, generando agrupaciones de mejor calidad que las generadas por otros algoritmos metaheurísticos como PSO (*Particle Swarm Optimization*).

8. CONCLUSIONES

En el documento se presentó una revisión de las investigaciones previas relacionadas con algoritmos metaheurísticos basados en la ley de gravitación universal aplicados a la tarea de agrupación de datos, se evidencio que es un campo poco explorado sin embargo que las investigaciones realizadas hasta el momento han entregado en todos los casos resultados que sobrepasan los resultados obtenidos por algoritmos convencionales de minería de datos como k-medias y otros algoritmos metaheurísticos como PSO (*Particle Swarm Optimization*), por otro lado que logran superar limitaciones como la vulnerabilidad a caer en óptimos locales que tiene los algoritmos convencionales populares como k-medias. Lo que lleva a decir que este tipo de propuestas poseen un gran potencial para el desarrollo de tareas de aplicación en diferentes problemas de agrupación de datos.

9. TRABAJOS FUTUROS

Se propone como trabajos futuros el desarrollo de estudios comparativos entre algoritmos metaheurísticos basados en la ley de gravitación universal y los algoritmos convencionales de minería de datos aplicados a diferentes problemas de agrupación de datos con el propósito de determinar en cuales problemas los primeros resuelven las limitaciones identificadas para los segundos.

10. REFERENCIAS

[1] F. Glover y G. Kochenberger, «Handbook of Metaheuristics.,» *Kluwer Academic Publishers,Norwell, MA,* 2012.

[2] E. Alba, «Parallel Metaheuiristics: A new class of algorithm,» *Wiley Interscience,,* 2005.

[3] I. H. Witten y E. Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second ed.), Burlington,USA: Morgan Kaufmann, 2005.

[4] E. Rashedi, H. Nezamabadi-pour y S. Saryzdi, «GSA: A Gravitational Search Algorithm.,» *Information Sciences,* vol. 179, pp. 2232-2248, 2009.

[5] A. Shafiq y D. R. P. Gillian, «An Evolutionary Particle Swarm Optimization Algorithm for Data Clustering,» de *IEEE Swarm Intelligence Symposium*, St. Louis MO USA, 2008.

[6] A. Hatamlou, S. Abdullah y H. Nezamabadi-pour, «A Combined Approach for Clustering Based on K-means and Gravitational Search Algorithms,» *Swarm and Evolutionary Computation,* nº 6, p. 47–52, 2012.

[7] J. Goméz, D. Dasgupta y O. Nasraoui, Compositores, *New Algorithm for Gravitational Clustering.* [Grabación de sonido]. In Proc. of the SIAM Int. Conf. on Data Mining. 2003.

[8] W. Wright, «Gravitational clustering,» *Pattern Recognition,* nº 9, pp. 151-166, 1977.

[9] S. Kundu, «Gravitational clustering: a new approach based on the spatial distribution of the points,» Computer Science Department, Louisiana State University, Baton Rouge, 1999.

[10] O. Castillo, J. Castro y A. Rodríguez, «Fuzzy granular gravitational clustering Algorithm,» *Fuzzy Information Processing Society Annual Meeting of the North American (NAFIPS),* 2012.

[11] A. Sadeghian y H. Nezamabadi-pour, «Gravitational Ensemble Clustering,» Department of Electrical Engineering, Shahid Bahonar University , Kerman, Irán, 2014.

[12] A. Hatamlou, S. Abdullah y Z. Othman, «Gravitational Search Algorithm with Heuristic Search for Clustering Problems,» *Data Mining and Optimization Conference On (DMO),* 2011.

[13] M. Dowlatshahi y H. Nezamabadi-pour, «GGSA:A Grouping Gravitational Search Algorithm for Data Clustering,» *Elservier - Engineering Applications of Artificial Intelligence,* vol. 36, pp. 114- 121, 2014.

[14] L. Peng, B. Yang, Y. Chen y A. Abraham, «Data gravitation based classification,» Information Sciences -Sciences Direct, vol. 179, nº 6, pp. 809-819, 2009.

[15] I. H. Witten y E. Frank, de Data Mining: Practical Machine Learning Tools and Techniques, Burlington, USA, Morgan Kaufmann, 2005.

[16] C. Pérez López, Minería de datos : técnicas y herramientas, Madrid - España: Thomson, 2007.

[17] H. Barrera, J. Correa y J. & Rodríguez, «Prototipo de Software para el preprocesamiento de datos "UD-Clear",» IV Simposio Internacional de Sistemas de Información e Ingeniería de Software en la Sociedad del Conocimiento SISOFT - Cartagena, Colombia, pp. 167-184, 2006.

[18] M. Berry y G. S. Linoff, Data Mining Techniques, Indianapolis, Indiana, USA: Wiley, 2004.

[19] J. E. R. Rodríguez, Fundamentos de la minería de datos, Bogotá D.C, Colombia : Universidad Distrital Francisco José de Caldas, 2010.

[20] V. J. Rayward-Smith, «Metaheuristics for Clustering in KDD,» Evolutionary Computation, 2005. The 2005 IEEE Congress, Australia, 2005.

[21] Y. Kuma y G. Sahoo, «A Review on Gravitational Search Algorithm and its Applications to Data Clustering & Classification,» Modern Education and Computer Science Press, nº 06, pp. 79-93, 2014.

[22] S. Abdullah, A. Hatamlou y Z. Othman, Compositores, Gravitational Search Algorithm with Heuristic Search for Clustering Problems. [Grabación de sonido]. Data Mining and Optimization Conference on (DMO). 2011.

[23] G. Oatley y B. Ewart, «Cluster Analysis and Data Mining Applications,» Analysis, vol. 1, nº 2, pp. 147-153, 2011.

[24] J. Hernandez, J. Ramírez y C. Ferri, Introducción a la Minería de datos, Editorial Alhambra S. A., 2004.

[25] T. Long y L. W. Jin, «A New Simplified Gravitational Clustering Method for Multi-prototype Learning Based on Minimum Classification Error Training Advances in Machine Vision, Image Processing, and Pattern Analysis, International Workshop on Intelligent Computing.,» Computer Science, N. Zheng, X. Jiang, and X. Lan, vol. 4153, pp. 168-175, 2006.