



Universidad Veracruzana
Instituto de Investigaciones en Inteligencia
Artificial

Implementación del Optimizador del Lobo Gris para el Minado de Reglas de Asociación

Requisito para obtener el grado de
Maestro en inteligencia artificial

Presenta:
Manuel Antonio Tiburcio Kuri

Director:
Dr. Horacio Tapia McClung

Xalapa, Veracruz, México

Mayo 2022

Agradecimientos

Me gustaría agradecer a las personas e instituciones que me apoyaron y contribuyeron para que pudiera realizar este trabajo de investigación. Primero quiero agradecer a mi familia, en especial a mis padres, los cuales siempre mostraron su confianza y apoyo incondicional para que pudiera cursar el posgrado, también deseo expresar mi agradecimiento al director de esta tesis, el doctor Horacio Tapia McClung, por la dedicación que ha brindado a este trabajo, al igual que su confianza durante el último año del programa, al Instituto de Investigaciones en Inteligencia Artificial por darme la oportunidad de ingresar en su posgrado de calidad y por último, pero no menos importante, al Consejo Nacional de Ciencia y Tecnología *CONACyT* por el apoyo económico sin el cual no hubiera sido posible la realización de esta tesis.

Resumen

El minado de reglas de asociación es una de las áreas más importantes de la minería de datos, esta técnica fue aplicada por primera vez para reconocer hábitos de compra de los clientes en un supermercado, a través una base de datos transaccional generada por un punto de venta *Point of Sales POS*. Esta área también ha sido utilizada en dominios como: análisis de crímenes, sector de la salud, ciberseguridad, entre otros. El objetivo principal de la Minería de Reglas de Asociación es encontrar asociaciones interesantes, con las diferentes combinaciones de artículos o ítems presentes en una base de datos, de manera que haya un antecedente que por ende tenga un consecuente.

Los enfoques tradicionales para el minado de reglas de asociación presentan un problema en común, estos generan una gran cantidad de reglas y para seleccionar un subconjunto entre ellas que aporte información valiosa y relevante, se vuelve una tarea computacionalmente costosa. En años recientes se ha propuesto el uso de algoritmos metaheurísticos y de optimización para solucionar esta problemática. En este trabajo de investigación se realiza un estudio de como el uso de la computación evolutiva, específicamente el campo de los algoritmos basados inteligencia de enjambre (*Swarm Intelligence SI*) forman parte del estado del arte en esta temática y se propone la implementación del optimizador del lobo gris binario mejorado *i-BGWO* para la generación de reglas de asociación.

Esta versión mejorada incorpora una nueva técnica llamada *DHL Dimension Learning-based Hunting*, la cual ha demostrado llegar a obtener mejores resultados en otros problemas de optimización. Con el fin de poder utilizar el optimizador del lobo gris para la generación de reglas de asociación, se convirtió al dominio binario. Con los experimentos realizados se encontró que el algoritmo propuesto puede llegar a obtener las mismas e incluso conjuntos de reglas de asociación con mayor calidad en comparación con otras propuestas del estado del arte. También se puede observar que la calidad de los resultados obtenidos por el algoritmo se mantiene a pesar de variar el número de objetos y transacciones existentes, generando reglas de interés para los diferentes casos de estudio, aunque se puede notar una desventaja con respecto el tiempo de ejecución en comparación con otras propuestas del estado del arte.

Dentro de la categoría de algoritmos en los que este trabajo se centra, el tipo más utilizado para esta tarea es el (*PSO Particle Swarm Optimization*) por sus siglas en ingles, representando el 15% del total las publicaciones relacionadas con la computación evolutiva para la minería de reglas de asociación *EARM*. Otros algoritmos basados en inteligencia de enjambre *other-SI-based*, solo representan el 1.8% de

las investigaciones referentes a esta temática. El Optimizador del Lobo Gris *GWO* pertenece a este grupo, haciéndolo un buen candidato para innovar en el área de *EARM*.

Tabla de contenidos

Índice

1. Introducción.	8
1.1. Objetivos.	10
1.1.1. Objetivos específicos.	10
1.2. Hipótesis	11
1.3. Estructura de la Tesis	11
2. Marco Teórico	12
2.1. Minería de datos	12
2.1.1. Metodología CRISP-DM	13
2.2. Reglas de asociación	15
2.2.1. Métricas para selección de reglas de asociación relevantes.	16
2.2.2. Clasificación de las reglas de asociación.	17
2.3. Minado de reglas de asociación en bases de datos transaccionales.	18
2.3.1. Representación de una base de datos transaccional.	18
2.3.2. Algoritmos para el minado de itemsets frecuentes.	19
2.4. Computación Evolutiva para la Minería de Reglas de Asociación.	22
2.4.1. Optimizador del Lobo Gris <i>GWO</i>	24
2.4.2. Optimizador del Lobo Gris Mejorado <i>i-GWO</i>	27
2.4.3. Optimizador del Lobo Gris Binario <i>BGWO</i>	30
3. Estado del Arte	33
3.1. Funciones de aptitud propuestas en la literatura.	37
4. Metodología	40
4.1. Etapa 1: Entendimiento del problema.	40
4.2. Etapa 2: Entendimiento de los datos.	40
4.3. Etapa 3: Preparación de los datos.	41
4.3.1. Base de Datos Mi Tiendita	42
4.3.2. Representación de las bases de datos transaccionales	48
4.4. Etapa 4: Modelado predictivo.	49
4.4.1. Representación de las reglas de asociación.	49
4.4.2. Función de aptitud <i>Fitness</i>	50
4.4.3. Algoritmo para el minar reglas de asociación <i>i-GBWO</i>	50

4.5. Etapa 5: Evaluación.	53
5. Resultados	54
5.1. Especificaciones técnicas de las pruebas realizadas.	54
5.2. Comparación de la calidad de las reglas generadas.	55
5.3. Caso de estudio enfocado al análisis de canasta de mercado con datos de un negocio local.	58
5.4. Prueba estadística Mann-Whitney-Wilcoxon.	60
5.5. Evaluación del tiempo de ejecución de <i>i-BGWO</i>	60
5.5.1. Por número de ítems.	61
5.5.2. Por número de transacciones.	62
6. Conclusiones y trabajo futuro.	63
7. Apéndices.	66

Índice de figuras

1.	Pasos de la metodología CRISP-DM [1].	14
2.	Representación de conjuntos de datos transaccionales [2]. . .	19
3.	Espacio de búsqueda para el conjunto de ítems $I = \{i_1, i_2, i_3, i_4\}$ [3]	20
4.	Jerarquía de los individuos en el liderazgo de la búsqueda de la solución más óptima.	24
5.	Distribución de las publicaciones relacionadas directamente a EARM, imagen tomada de [4].	34
6.	Taxonomía de los algoritmos evolutivos usados en la minería evolutiva de reglas de asociación EARM, imagen tomada de [4].	35
7.	Distribución de los diferentes dominios donde se utiliza la minería evolutiva de reglas de asociación EARM[4].	36
8.	Método de búsqueda vertical para contar la frecuencia de cada ítem [5].	42
9.	Serie de tiempo con el costo de cada transacción por mes. . .	46
10.	Ventas de enero 2016.	46
11.	Gráfico Pareto de los 20 productos que más generan ingreso. .	47
12.	Metodología implementada para la programación del algorit- mo i -BGWO.	52
13.	Tiempo de ejecución por cantidad de ítems.	61
14.	Tiempo de ejecución por cantidad de transacciones.	62

Índice de tablas

1.	Base de datos transaccional.	16
2.	Conjuntos de datos utilizados.	40
3.	Archivos origen de base de datos (Mi Tiendita).	42
4.	Descripción general de los atributos.	43
5.	Descripción de tipo de atributos.	43
6.	Descripción general de los atributos.	44
7.	Estadísticas descriptivas de la base de datos.	44
8.	Cantidad de productos por transacción.	45
9.	Base de datos transaccional.	48
10.	Base de datos transaccional binarizada.	48
11.	Valor de aptitud promedio de las reglas obtenidas por los algoritmos.	55
12.	Estadísticas descriptivas para los promedios de los valores de aptitud obtenidos por los algoritmos implementados.	56
13.	Valor de confianza promedio de las reglas obtenidas por los algoritmos.	57
14.	Estadísticas descriptivas para los promedios de los valores de confianza obtenidos por los algoritmos implementados.	57
15.	Promedio de la función de aptitud y confianza de las reglas generadas con la base de datos Mi Tiendita.	59
16.	Estadísticas descriptivas para los promedios de los valores de aptitud y confianza obtenidos por la base de datos de Mi Tiendita.	59
17.	Valores de p obtenidos con la prueba Mann-Whitney-Wilcoxon.	60
18.	10 mejores reglas de asociación generadas, correspondientes al primer trimestre de 2016.	66
19.	10 mejores reglas de asociación generadas, correspondientes al segundo trimestre de 2016.	67
20.	10 mejores reglas de asociación generadas, correspondientes al tercer trimestre de 2016.	68
21.	10 mejores reglas de asociación generadas, correspondientes al cuarto trimestre de 2016.	69

1. Introducción.

El objetivo principal de la minería de datos *MD* es extraer de manera automática información valiosa y no fácilmente reconocible en bases de datos extensas [6], de esta manera ayudar a responder preguntas que suelen demandar mucho tiempo de los analistas, permitiendo la toma de decisiones basada en conocimiento fundamentado y confiable a partir de la información disponible. En la actualidad su uso ha ido en aumento debido al crecimiento exponencial de los datos y la necesidad por extraer valor de ellos. Dependiendo del problema de estudio, la minería de datos se puede plantear desde dos perspectivas: desde el punto de vista descriptivo o desde el punto de vista predictivo [7].

La presente investigación cuenta con un enfoque descriptivo, debido a que se utiliza la técnica de minería de reglas de asociación *ARM* (Association Rule Mining por sus siglas en inglés)[8] para la generación de asociaciones entre los ítems existentes en bases de datos transaccionales. La minería de reglas de asociación *ARM* es un área la cual ha ganado interés en la comunidad en los últimos años, esta ha sido utilizada en diferentes problemas del mundo real, como reconocer los hábitos de compra de los clientes, en el campo de la medicina, detección de crímenes, entre otros [9]. El objetivo principal de la Minería de Reglas de Asociación *ARM* es encontrar asociaciones de interés en conjuntos de datos transaccionales, de manera que los hechos que existen en una base de datos se toman como antecedentes que por ende traigan consigo consecuentes, formando diversas combinaciones para crear conjuntos de ítems [10].

Los enfoques tradicionales del minado de reglas de asociación, utilizan estrategias convencionales para afrontar el problema de la generación de conjuntos de ítems frecuentes como, divide y vencerás o fuerza bruta. Estas estrategias generan un gran número de reglas, por lo que discriminar entre ellas conjuntos de ítems relevantes resulta ser computacionalmente costoso [4]. Muchos estudios se han concentrado en el desarrollo de algoritmos para la extracción de conjuntos de datos (*itemsets*) frecuentes, debido a que es la tarea más compleja en el proceso de extracción de reglas de asociación [10].

Agrawal et al. [8] fueron los pioneros en proponer un algoritmo para extraer reglas de asociación de bases de datos transaccionales, al cual llamaron *Algoritmo Apriori*. Este algoritmo presenta la debilidad de que realiza varios recorridos a la base de datos para poder contar la repetición de los candi-

datos a *conjuntos de datos* frecuentes. Después de Apriori se desarrollaron otros algoritmos para el minado de reglas de asociación, siendo FP-Growth [11] uno de los más relevantes, debido a la superioridad en tiempo de ejecución en comparación con Apriori. La ventaja que implementa este algoritmo es que construye una estructura compacta llamada FP-Tree, requiriendo solo dos recorridos a la base de datos, aunque la ejecución y memoria se ven comprometidas cuando las bases de datos contienen una vasta cantidad de transacciones e ítems.

El uso de algoritmos evolutivos para minar de reglas de asociación, permite generar estas haciendo uso de la optimización y sin tener que realizar el paso de generar *conjuntos de datos* frecuentes, lo que trae consigo una reducción considerable en el tiempo de ejecución [12]. En las últimas dos décadas se han presentado modelos para el descubrimiento de reglas de asociación basado en este tipo de algoritmos metaheurísticos, enfrentando las limitaciones que presentan los enfoques tradicionales. A este campo se le conoce como Computación Evolutiva para la Minería de Reglas de Asociación (ECARM Evolutionary Computation for Association Rule Mining) [4].

En la mayoría de los trabajos en el campo de ECARM solo se han utilizado los algoritmos más populares del área, siendo estos los Algoritmos Genéticos *GAs* y *Particle Swarm Optimization PSO* [4], el último formando parte de los modelos basados en inteligencia de enjambres *Swarm Intelligence SI*. Dentro de estos, se encuentra el optimizador del lobo gris *GWO* propuesto por S. Mirjalili et al. [13] en 2014. Este algoritmo imita la jerarquía de liderazgo y el mecanismo de caza de los lobos grises en la naturaleza. Al ser una propuesta reciente, se han realizado pocas investigaciones en el minado de reglas de asociación con este algoritmo, lo que le convierte en un buen candidato para la exploración de nuevas oportunidades de innovación en el área de la ECARM.

En este trabajo de tesis se propone una modificación de la versión mejorada del optimizador del lobo gris *i-BGWO* expuesta en [14], a partir de la propuesta presentada en [13], donde se modifica la estructura del algoritmo para procesar información binaria, con el objetivo de generar reglas de asociación provenientes de bases de datos transaccionales. Esta versión binaria lleva por nombre: Optimizador Mejorado del Lobo Gris Binario *i-BGWO*. El lobo gris mejorado *i-GWO* incrementa considerablemente el desempeño del optimizador, atacando los problemas de falta de diversidad en la población,

exploración y convergencia prematura en el lobo gris canónico, implementando una nueva estrategia de movimiento llamada *dimension learning-based hunting DLH*, esta estrategia de búsqueda se basa en el comportamiento de caza individual de los lobos en la naturaleza. *DLH* construye un vecindario de donde cada lobo aprende de sus vecinos, compartiendo la información disponible entre ellos.

La versión discreta del lobo gris *i-BGWO* arroja resultados competitivos en comparación con otros modelos de la computación evolutiva para la minería de reglas de asociación. Con los experimentos realizados siendo un algoritmo capaz de igualar y superar el valor de la función de aptitud y la calidad de las reglas de asociación generadas en comparación con otros modelos para el minado de reglas de asociación.

1.1. Objetivos.

- Implementar el optimizador mejorado del lobo gris *i-BGWO* y modificarlo para usarlo en el dominio binario, con el fin de generar reglas de asociación las cuales permitan predecir de manera acertada las relaciones entre dos o más ítems de una base de datos transaccional, al mismo tiempo que comparar los resultados obtenidos con otros algoritmos del estado del arte en la Computación Evolutiva para la Minería de Reglas de Asociación.

1.1.1. Objetivos específicos.

- Revisar la literatura sobre la minería de reglas de asociación.
- Implementar y probar el *i-BGWO*, comparándolo con otros algoritmos pertenecientes a la computación evolutiva para la generación de reglas de asociación.
- Entender la representación de las reglas de asociación en bases de datos transaccionales.
- Implementar cambios en los algoritmos evolutivos para que puedan procesar bases de datos transaccionales y generar reglas de asociación.

1.2. Hipótesis

Al modificar el algoritmo para usarlo en el dominio binario, el optimizador lobo gris mejorado permitirá obtener conjuntos de reglas de asociación de calidad, que posean métricas de aptitud y confianza competitivas y en algunos casos superiores contra otras alternativas del estado del arte para la generación de reglas de asociación, midiendo los resultados cuantitativos obtenidos contra los expuestos en la literatura.

1.3. Estructura de la Tesis

Este trabajo está estructurado de la siguiente manera: el primer capítulo es una introducción general al problema, donde se presentan los objetivos de la investigación. En el segundo capítulo, se proporcionan los conceptos y el marco teórico que sirven como fundamento para el desarrollo del proyecto, y proporciona un entendimiento más amplio sobre el tema de estudio presentado. El tercer capítulo, describe la justificación de la tesis, al igual que el estado del arte y trabajos relacionados que fungen como base para el desarrollo de este trabajo de investigación. En el cuarto capítulo se presenta una descripción detallada acerca de la estrategia de solución propuesta. El quinto capítulo, detalla la implementación, la experimentación realizada y los resultados obtenidos de ellos. En el sexto y último capítulo se exponen las conclusiones a partir de los resultados, se discuten y se realizan propuestas de trabajo futuro.

2. Marco Teórico

Este capítulo se detalla la teoría en la que se basa este proyecto de investigación, primero se describe que es la minería de datos y la metodología CRISP-DM, seguido se explica que es la minería de reglas de asociación y su clasificación y por último se ahonda en el campo de la minería evolutiva de reglas de asociación específicamente en los modelos basados en inteligencia de enjambre y como estas meta-heurísticas son utilizadas para el minado de reglas de asociación.

2.1. Minería de datos

Charu C. Aggarwal, describe la minería de datos como el estudio de la recopilación, limpieza, procesamiento, análisis y obtención de información útil y comprensible, previamente desconocido, a partir de datos almacenados en distintos formatos [6]. El flujo de trabajo de una aplicación típica de minería de datos contiene las siguientes fases:

- Recopilación de datos: la recopilación de datos puede requerir el uso de hardware especializado, como una red de sensores, trabajo manual, también con herramientas de software como la recopilación de encuestas a usuarios, o formularios para recopilar información.
- Extracción de características y limpieza de datos: cuando se recopilan los datos, a menudo no están en una forma adecuada para su procesamiento. Por ejemplo, los datos pueden estar codificados en registros complejos o documentos de formato libre. En muchos casos, diferentes tipos de datos pueden mezclarse arbitrariamente. El resultado final de este procedimiento es un conjunto de datos muy bien estructurado, que puede ser utilizado de manera efectiva por un programa de computadora. Después de la fase de extracción de características, los datos pueden volver a almacenarse en una base de datos para su procesamiento.
- Procesamiento analítico y algoritmos: la parte final del proceso de minería es diseñar métodos analíticos efectivos a partir de los datos procesados.

Llevar a cabo el proceso de la minería de datos requiere la aplicación de una metodología estructurada para su correcta utilización, en la literatura existen tres [15]. KDD (Knowledge Discovery in Databases), SEMMA

(Sample, Explore, Modify, Model, Assess) y CRISP-DM (Cross Industry Standard Process for Data Mining). Específicamente para esta investigación y por el tipo de problema de estudio se utilizará la metodología CRISP-DM debido a su gran adaptabilidad. Con respecto a la última etapa de la metodología CRISP-DM solo se considerará el paso de la creación del reporte final, el cual está constituido por la escritura de este documento.

2.1.1. Metodología CRISP-DM

La metodología CRISP-DM (Cross Industry Standard Process for Data Mining) es considerada la guía de referencia más popular en el desarrollo de proyectos de minería de datos, por su capacidad de adaptabilidad y detalle [1]. Con esta metodología se irá creando un flujo de procesos personalizado que se vaya ajustando a las necesidades del proyecto. Esta flexibilidad es una de las razones por la que se optó por esta metodología, la cual consta de los siguientes pasos:

- Entendimiento del problema.
 - Determinación de los objetivos del negocio.
 - Valoración de la situación.
 - Determinación de los objetivos de minería de datos.
 - Producción de un plan de proyecto.
- Entendimiento de los datos
 - Recopilación de datos iniciales.
 - Descripción de los datos.
 - Análisis exploratorio de los datos.
 - Verificación de calidad de los datos
- Preparación de datos:
 - Selección de los datos
 - Limpieza de los datos
 - Construcción de los datos
 - Integración de los datos
 - Formato de los datos
- Modelado predictivo:

- Selección de técnicas de modelado
 - Generación del modelo
 - Evaluación del modelo
- Evaluación:
 - Evaluación de los resultados
 - Proceso de revisión
 - Determinar pasos siguientes
 - Distribución:
 - Planificación de control y mantenimiento.
 - Creación de informe final
 - Revisión del proyecto

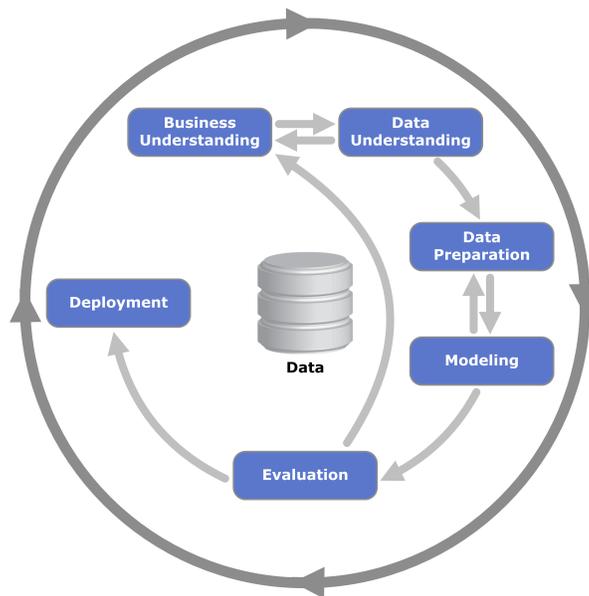


Figura 1: Pasos de la metodología CRISP-DM [1].

2.2. Reglas de asociación

Las Reglas de Asociación son utilizadas para descubrir asociaciones y hechos interesantes que ocurren en común dentro de un determinado conjunto de datos. Los algoritmos para generar estas reglas son generalmente aplicados a bases de datos transaccionales. Estas bases de datos se refieren a un conjunto de registros de transacciones en los que cada transacción está compuesta por un conjunto de ítems [16].

Un ejemplo de una base de datos transaccional puede ser la que se genera en un punto de venta (POS, point of sales) de una tienda de conveniencia, donde no hay ningún atributo que especifique el número de identificación de los compradores, solo el número identificador de cada venta que se realiza junto con el conjunto de ítems comprados. La técnica para minar estas reglas de asociación fue por primera vez propuesta en este contexto, descubrir patrones y relaciones entre los productos que se vendían juntos en un supermercado en 1993 [8].

Una regla de asociación está compuesta por un antecedente y un consecuente, los cuales conforman un conjunto de ítems denominado *itemset*. Formalmente, $I = \{i_1, i_2, \dots, i_n\}$ representa un conjunto de n atributos binarios llamados ítems y $D = \{t_1, t_2, \dots, t_n\}$ representa un conjunto de transacciones almacenadas en una base de datos, cada transacción en D tiene un *Id* (identificador único) y este contiene un subconjunto de ítems de I (denominado *itemset*). De este modo, una regla de asociación se puede definir de la siguiente manera: si X entonces Y , ($X \Rightarrow Y$), donde el antecedente lo representa X y el consecuente Y . X y Y son subconjuntos de I , ($X, Y \subseteq I$), estos conforman un conjunto de ítems (*itemset*) donde su intersección debe de ser vacía ($X \cap Y = \emptyset$), es decir sin ítems en común [6]. Por ejemplo en la Tabla 1, la cual contiene 5 transacciones, se pueden obtener las siguientes reglas de asociación hipotéticas: ($Pan, Leche \Rightarrow Huevo$), ($Pan, Cerveza \Rightarrow Pañales$).

Transacciones	
ID	Artículos
1	Queso, Pan, Jamón
2	Pan, Pañales, Cerveza
3	Leche, Cerveza, Pañales
4	Pan, Queso, Huevo
5	Huevo, Pañales, Cerveza, Pan

Tabla 1: Base de datos transaccional.

2.2.1. Métricas para selección de reglas de asociación relevantes.

Un problema que presenta minar reglas de asociación es poder discernir entre las reglas que realmente aportan información valiosa del conjunto de transacciones. Las métricas más utilizadas para ayudar a entender la fuerza de asociación que existe entre los conjuntos de antecedentes y consecuentes son: el soporte, la confianza y el levantamiento. Ellas describen la calidad, probabilidad, generalidad y fiabilidad de una regla.

El soporte se refiere a la popularidad predeterminada de un ítem o conjuntos de ítems en la base de datos, esta métrica se calcula buscando el número de transacciones que contienen el ítem o el conjunto de ítems, entre el número de total de transacciones.

$$\text{Soporte}(X) = \frac{\text{Transacciones que contienen } X}{\text{Total de transacciones}} \quad (1)$$

$$\text{Soporte}(X \Rightarrow Y) = \frac{\text{Transacciones que contienen}(X \Rightarrow Y)}{\text{Total de transacciones}} \quad (2)$$

La Confianza se refiere a la probabilidad de que si el ítem X es comprado, también lo será el ítem Y. Esta métrica se calcula encontrando el soporte de $(X \Rightarrow Y)$, dividido por el soporte de (X) .

$$\text{Confianza}(X \Rightarrow Y) = \frac{\text{Transacciones que contienen}(X \Rightarrow Y)}{\text{Transacciones que contienen}(X)} \quad (3)$$

El levantamiento indica el incremento en la proporción de venta de X cuando se vende Y, explicando la probabilidad de que comprar X y Y juntos es más probable que solo comprar X. Un levantamiento mayor a 1, indica que será

más probable que los productos X y Y se compren juntos. Un levantamiento igual a 1, significa que no hay una asociación entre los productos y un levantamiento de menor a 1 significa que es poco probable que X y Y se compren juntos. Este es calculado dividiendo la confianza ($X \Rightarrow Y$) entre el soporte $Soporte(B)$.

$$Lift(X \Rightarrow Y) = \frac{Confianza(X \Rightarrow Y)}{Soporte(Y)} \quad (4)$$

2.2.2. Clasificación de las reglas de asociación.

Las reglas de asociación pueden ser clasificadas acorde a los siguientes criterios [17].

1. Según los tipos de valores que manejan las reglas:

- Reglas de asociación booleanas: estas son expresadas entre la presencia o ausencia de un ítem. Son utilizadas en la formación de conjuntos de datos *item-sets*.

Regla 1: (Si compra teléfono móvil \Rightarrow Compra funda protectora).

- Reglas de asociación cuantitativas: Estas describen asociaciones entre ítems y atributos cuantitativos divididos en intervalos.

Regla 2: (edad ($X, \ll 25 \dots 35 \gg$) \wedge ingreso ($X, \ll 100,000 \dots 300,000 \gg$) \Rightarrow compra ($X, \text{teléfono móvil}$)).

La regla 2 indica que hay una relación entre la edad del cliente X, sus ingresos y el teléfono móvil X que este cliente compra.

2. Según las dimensiones de datos que involucra una regla.

- Reglas de asociación de dimensión simple: Los atributos de una regla de asociación se referencian solo en una dimensión, la regla 1 y 3 se pueden tomar como reglas de dimensión simple, debido a que corresponden al hecho de aparecer o no en la transacción.

Regla 3: (Si compra pan \Rightarrow Compra leche).

- Reglas de asociación multidimensionales: Los atributos de una regla de asociación se referencian solo en dos o más dimensiones, por ejemplo, tomando en cuenta el tiempo en la regla tres.

Regla 4: (Si Cliente (X) compra pan en Tiempo(Mayo) \Rightarrow Compra leche).

3. Con base en los niveles de abstracción.

- Estos permiten incorporar a las reglas diferentes niveles de abstracción representados por conceptos que aglutinan otros conceptos, por ejemplo:

Regla 5: (edad ($X, \ll 25 \dots 35 \gg$) \Rightarrow compra (X , teléfono móvil)).

Regla 6: (edad ($X, \ll 25 \dots 35 \gg$) \Rightarrow compra (X , teléfono)).

4. Instantáneas o secuenciales.

- Estas reglas expresan patrones de comportamientos secuenciales que se dan en instantes distintos pero cercanos de tiempo. Por ejemplo, si un cliente compra un cepillo y pasta de dientes, probablemente comprará hilo dental la próxima vez que vaya a comprar.

2.3. Minado de reglas de asociación en bases de datos transaccionales.

En esta sección se muestran las formas en que se pueden representar los conjuntos de datos transaccionales, al igual que los algoritmos más utilizados para minar reglas de asociación en este tipo de bases de datos.

2.3.1. Representación de una base de datos transaccional.

Existen tres formas de representar una base de datos transaccional. [2]

- Representación binaria (Figura 2a): cada transacción es almacenada en un vector binario donde un 1 denota la presencia de un ítem y 0 la ausencia del mismo.
- Representación horizontal (Figura 2b): consiste en un conjunto de pares (r, cas), donde r es el identificador de la transacción y cas es una secuencia de *conjuntos de datos*. Esta representación almacena por cada transacción una lista con los ítems presentes en ella.

- Representación vertical (Figura 2c): consiste en un conjunto de pares (a, lista) donde a es un ítem y lista es una secuencia ordenada de los identificadores de las transacciones donde el ítem está presente.

	a b c d e		a b c d e
1	1 1 0 1 1	1 → abde	↓ ↓ ↓ ↓ ↓
2	0 1 1 0 1	2 → bce	1 1 2 1 1
3	1 1 0 1 1	3 → abde	3 2 4 3 2
4	1 1 1 0 1	4 → abce	4 3 5 5 3
5	1 1 1 1 1	5 → abcde	5 4 6 6 4
6	0 1 1 1 0	6 → bcd	5 5
			6

(a) Booleana

(b) Horizontal

(c) Vertical

Figura 2: Representación de conjuntos de datos transaccionales [2].

El proceso de minado de reglas de asociación consta de dos pasos fundamentales:

- La búsqueda de los *conjuntos de datos* frecuentes en el conjunto de datos.
- La generación de las reglas de asociación a partir de los *conjuntos de datos* frecuentes. interesantes.

2.3.2. Algoritmos para el minado de itemsets frecuentes.

La generación de conjuntos de datos frecuentes es el paso más costoso del minado de reglas de asociación. Una gran parte de los trabajos existentes en se enfocan en mejorar la eficiencia de este proceso. Esta complejidad se debe al tamaño del espacio de búsqueda, el cual consiste en la cantidad de transacciones e ítems en la base de datos. Para afrontar el problema de recorrer todo el espacio de búsqueda, Apriori emplea el principio denominado Clausura Descendente del Soporte [8]. Este principio plantea que todo subconjunto de un (*itemset*) frecuente es frecuente, mientras que todos los subconjuntos de (*conjuntos de datos*) no frecuentes, no lo son. Esto dictado por la métrica del soporte, donde los conjuntos de datos que no pasan el umbral mínimo son descartados, formando así una frontera que divide el espacio de búsqueda en dos subespacios, El que contiene los conjuntos frecuentes y el que no.

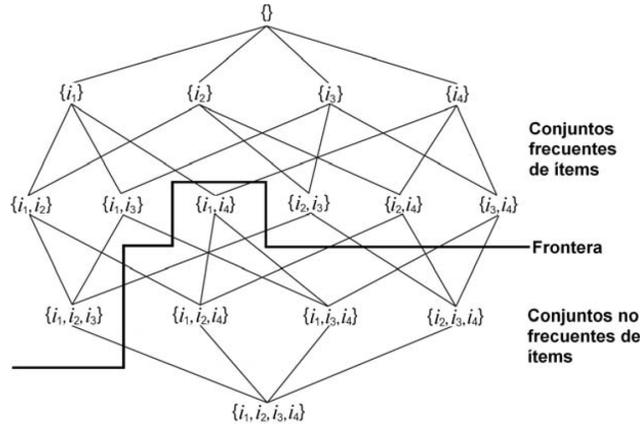


Figura 3: Espacio de búsqueda para el conjunto de ítems $I = \{i_1, i_2, i_3, i_4\}$ [3]

La figura 3 [3] muestra la frontera en el espacio de búsqueda para el conjunto de ítems $I = \{i_1, i_2, i_3, i_4\}$ para el conjunto de transacciones $T = \{\{i_1, i_2, i_3\}, \{i_2, i_4\}, \{i_3, i_4\}, \{i_1, i_2, i_3, i_4\}\}$ y dado un umbral de soporte mínimo de $minSupp = 0.5$ (considerado que los itemsets son frecuentes si aparecen por lo menos en el 50% de las transacciones) [3].

Los itemsets frecuentes pueden ser generados de dos formas:

- En amplitud: Se generan todos los conjuntos frecuentes de ítems de tamaño k antes de generar los conjuntos de ítems de tamaño $(k+1)$ Apriori [8], AprioriTid [18], Apriori-Hybrid [18], DHP [19], IHP [20].
- En profundidad: Recursivamente se generan los conjuntos ítems por cada rama de la estructura arbórea FP-Growth [11], Patricia Trie-Mine [21], CT-ITL [22], CT-PRO [23].

Apriori es el algoritmo pionero para el minado de *conjuntos de datos* frecuentes, Él cual identifica de manera eficiente la frecuencia de conjuntos de ítems y con estos poder extraer todos los *conjuntos de datos* frecuentes que superan el umbral mínimo de soporte. El inconveniente que presenta el algoritmo Apriori, es la cantidad exponencial del tiempo de ejecución y de memoria al generar generar todos los conjuntos de candidatos. Lo anterior se debe a que recorre la base de datos varias veces para buscar todos los conjuntos de ítems frecuentes, al mismo tiempo que el número de subconjuntos de los *conjuntos de datos* frecuentes candidatos esta en continuo aumento.

Después de Apriori, se desarrollaron otras opciones para la minería de *conjuntos de datos* frecuentes con el objetivo hacer más eficiente el proceso, uno de los más representativos es el FP-Growth [11], el cual en lugar de utilizar un enfoque de fuerza bruta hace uso del principio divide y vencerás, generando recursivamente los conjuntos frecuentes de ítems. Este algoritmo utiliza una estructura arbórea llamada *FP-Tree* (Frequent Pattern Tree) para generar los conjuntos de ítems frecuentes. La ventaja principal de FP Growth es que escanea la base de datos solo dos veces y no genera tantos conjuntos de candidatos, este algoritmo consta principalmente de dos etapas:

1. Construcción del árbol *FP-Tree*, recorriendo el conjunto de transacciones en busca de los itemsets frecuentes y contabilizar sus respectivas frecuencias. Se ordenan los ítems de manera descendente acorde a sus frecuencias, de esta manera los ítems con mayor frecuencia estarán más cercanos a la raíz del árbol, seguidamente se recorre el conjunto de transacciones, solo con los *conjuntos de datos* frecuentes ya ordenados, para componer cada rama del árbol.
2. Poda del árbol para obtener los *conjuntos de datos* frecuentes.

2.4. Computación Evolutiva para la Minería de Reglas de Asociación.

La computación evolutiva forma parte de la inteligencia artificial, esta área está conformada por diversos grupos de algoritmos de optimización que se inspiran en la evolución biológica, al tener un carácter estocástico y meta-heurístico, pueden encontrar posibles soluciones óptimas para una gran cantidad de problemas de optimización, evaluando la aptitud de los individuos con una función de aptitud *Fitness Function* [24], por esta razón, se le considera parte del estado del arte [4].

El uso de algoritmos evolutivos para la minería de reglas de asociación, permite descubrir reglas de asociación sin tener que hacer el paso de la generación de conjuntos de ítems frecuentes, lo que trae consigo una reducción considerable en el tiempo de ejecución [12]. En las últimas dos décadas, investigadores han presentado modelos para el descubrimiento de reglas de asociación basado en este tipo de algoritmos metaheurísticos, enfrentando las limitaciones que presentan los enfoques tradicionales [4]. Estos algoritmos han demostrado ser de gran utilidad en el minado de reglas de asociación, los algoritmos metaheurísticos más utilizados en el área para esta tarea son:

1. Enjambre de partículas: Este algoritmo está inspirado en el comportamiento social del vuelo de las parvadas. Utiliza se utiliza una población de tamaño fijo donde, cada partícula de la población es una posible solución al problema y sus movimientos se encuentran acotados al espacio de búsqueda, este forma parte de los algoritmos basados en inteligencia de enjambres y es el más utilizado de esta área hasta el momento en la minería de reglas de asociación. [4, 25, 26, 5].
2. Colonia de Hormigas: Este algoritmo imita el comportamiento natural de las hormigas en la búsqueda de alimento. Cuando éstas encuentran alimento (solución candidata al problema), evalúan la fuente y dejan un rastro de feromonas, cuya cantidad depende de la calidad del alimento (calidad de la solución encontrada) y la distancia con respecto a éste. Si las hormigas no detectan dicha sustancia se mueven de forma aleatoria, pero si la perciben es muy probable que se desplacen siguiendo el rastro, lo que a su vez aumenta la cantidad de las feromonas. El depósito de esta sustancia guía a otras hormigas hacia la fuente de alimento [27, 28].
3. Enjambre de Abejas: Este algoritmo de inteligencia de enjambres se

inspira en el comportamiento de las abejas en la búsqueda de néctar. Las abejas se mueven en un espacio de búsqueda multidimensional eligiendo fuentes de néctar (soluciones candidatas al problema) dependiendo de su experiencia pasada y la de sus compañeros de colmena o ajustando su posición. Algunas abejas (exploradoras) vuelan y eligen las fuentes de alimento aleatoriamente sin usar experiencia. Cuando encuentran una fuente de néctar mayor (una solución de mejor calidad), memorizan su posición y olvidan la anterior. De este modo, el algoritmo combina métodos de búsqueda local y global, equilibrando el proceso de la exploración y explotación del espacio de búsqueda [29, 30].

4. Algoritmos Genéticos: Este algoritmo es una analogía del proceso de adaptación de los seres vivos en un ambiente biológico, cada solución es un individuo de una población y se codifica como un vector binario. En cada una de las iteraciones, se aplican operadores evolutivos (selección, cruzamiento y mutación) los cuales combinan y modifican a los individuos de la población para crear una nueva población de soluciones [31, 32].
5. Optimizador del lobo gris: Es un algoritmo que imita la jerarquía de liderazgo y el mecanismo de caza de los lobos grises en la naturaleza. Tiene como objetivo encontrar un posible óptimo global de una función a través de la optimización de la posición de los tres mejores individuos (lobos) de la manada en cada iteración [13]. Este optimizador ha demostrado igualar y a veces llegar a obtener mejores resultados que los algoritmos previamente mencionados, por ello en el siguiente capítulo será analizado y comparado con PSO. En la siguiente sección se detallan los pasos específicos de este algoritmo.

2.4.1. Optimizador del Lobo Gris *GWO*

El algoritmo *GWO* tiene como objetivo encontrar un posible óptimo global de una función a través de la optimización de la posición de los tres mejores individuos (lobos) de la manada en cada iteración [13]. Se emplean cuatro tipos de lobos grises para simular la jerarquía de liderazgo: alfa, beta, delta y omega (figura 4), donde los primeros tres conforman a las posiciones de un individuo, mientras que los demás son considerados omegas. Este optimizador cuenta con tres etapas fundamentales: búsqueda, acorralamiento y ataque.

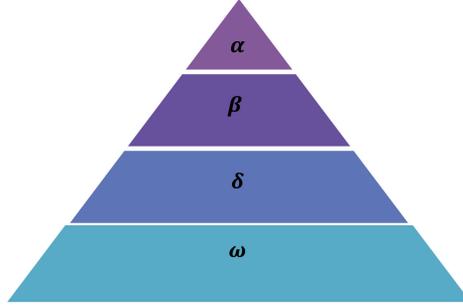


Figura 4: Jerarquía de los individuos en el liderazgo de la búsqueda de la solución más óptima.

Para que la manada pueda cazar una presa, primero es rodeada. Para modelar matemáticamente este comportamiento, se utilizan las siguientes ecuaciones.

$$\vec{X}(t+1) = \vec{X}_p(t) + \vec{A} \cdot \vec{D} \quad (5)$$

Donde \vec{D} es como se define en la ecuación (6), t es el número de iteración, \vec{A} y \vec{C} son vectores de coeficientes, \vec{X}_p es la posición de la presa y \vec{X} es la posición del lobo gris.

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right| \quad (6)$$

Los vectores A y C se calculan en las ecuaciones (7) y (8).

$$\vec{A} = 2a \cdot \vec{r}_1 - a \quad (7)$$

$$\vec{C} = 2\vec{r}_2 \quad (8)$$

donde a disminuye linealmente de 2 a 0 acorde a la ecuación 16 en el transcurso de las iteraciones debido a la naturaleza del problema plasmado en [13], y $r1$, $r2$ son vectores aleatorios con un rango de $[0, 1]$. El comportamiento de caza de los lobos grises se representa de la siguiente manera, el lobo Alfa es la mejor solución, el cual lleva el liderazgo, la segunda mejor solución candidata es Beta y la tercera delta, estos tres lobos son los que tienen mejor conocimiento sobre la ubicación potencial de la presa. Las tres primeras mejores soluciones candidatas obtenidas hasta el momento, obligan a los demás lobos (Omegas) en el espacio de búsqueda a actualizar sus posiciones según la posición de los tres mejores (Ecuación 9). $X_{i-GWO}(t + 1)$

$$\vec{X}_{i-GWO}(t + 1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (9)$$

donde, $\vec{X}_1, \vec{X}_2, \vec{X}_3$ están definidos en las ecuaciones (10), (11), (12) respectivamente.

$$\vec{X}_1 = \left| \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \right|, \quad (10)$$

$$\vec{X}_2 = \left| \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \right|, \quad (11)$$

$$\vec{X}_3 = \left| \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \right|, \quad (12)$$

donde, $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_{delta}$ son las primeras tres mejores soluciones en la manda en la iteración t , $\rightarrow A1$, $\rightarrow A3$, $\rightarrow A3$, están definidos en la ecuación 7 y $\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_{delta}$, están definidas en las ecuaciones (13), (14) y (15) respectivamente.

$$\vec{D}_\alpha = \left| \vec{C}_1 - \vec{X} \cdot \vec{X} \right|, \quad (13)$$

$$\vec{D}_\beta = \left| \vec{C}_2 - \vec{X}_\beta \cdot \vec{X} \right|, \quad (14)$$

$$\vec{D}_\delta = \left| \vec{C}_3 - \vec{X}_\delta \cdot \vec{X} \right|, \quad (15)$$

donde, $\rightarrow C_1, \rightarrow C_2$ y $\rightarrow C_3$ están definidas en la ecuación 8.

Una observación final sobre el optimizador de lobo gris (GWO) es la actualización del parámetro que controla el equilibrio entre la exploración y

caza. El parámetro a se actualiza linealmente en cada iteración donde varía de 2 a 0 según la ecuación 16.

$$a = 2 - t \frac{2}{MaxIter}, \quad (16)$$

donde, t es el número de la iteración y $MaxIter$ es el total de iteraciones permitidas en el parámetro del algoritmo, (Algoritmo 1).

Algorithm 1: El Optimizador del Lobo Gris (GWO)

Input: $N, D, Maxiter$

Output: Posible Óptimo Global

```

1 begin
2   ┌ Inicialización (Distribución aleatoria de  $N$  lobos en el espacio de
   └ búsqueda y calculo de su función de aptitud.
3 for  $iter = 2$  to  $Maxiter$  do
4   ┌ Encuentra  $X_\alpha, X_\beta$  and  $X_\delta$ .
5   ┌ for  $i = 1$  to  $N$  do
6   ┌ ┌ Computar  $X_{i-GWO}(t + 1)$ 
7   ┌ ┌ Selección del mejor ( $X_{i-GWO}(t + 1), X_i(t + 1)$ ).
8   ┌ ┌ Actualizar la población.
9 return El optimo global

```

Los mejores tres lobos, siempre guían a los omegas durante la búsqueda y caza hacia el área más prometedora para encontrar una solución competitiva. Cuando se encuentra la presa, esta es acorralada y luego atacada, sin embargo, este comportamiento puede hacer que el algoritmo sufra de falta de diversidad en la población, lo que puede llevarlo a quedar atrapado en un mínimo local y en consecuencia a tener una convergencia prematura.

2.4.2. Optimizador del Lobo Gris Mejorado *i-GWO*

Para afrontar el problema anteriormente mencionado, se propuso en la literatura una versión mejorada, llamada *improved-GWO* presentada en [14], donde se implementa una nueva estrategia de búsqueda asociada a la selección y actualización de los lobos, denominada (*DLH Dimension learning-based hunting*). Esta se inspira en el comportamiento de caza individual de los lobos grises, donde a partir de cada individuo se crea un vecindario, donde aprende de sus vecinos para poder generar un nuevo candidato, de esta manera compararlos y elegir cual debe ser acreedor a tomar la posición del alfa.

A continuación se describen detalladamente las tres etapas del optimizador del lobo gris mejorado *i-GWO*: Búsqueda (inicialización), movimiento (acorrallamiento) y ataque (selección y actualización).

Etapas de inicialización: En esta etapa, n lobos son generados aleatoriamente en el espacio de búsqueda dado por un rango, (Ecuación 17).

$$X_{ij} = l_j + \text{rand}_j[0, 1] \times (u_j - l_j), i \in [1, N], j \in [1, D] \quad (17)$$

Etapas de movimiento: *i-GWO* incorpora un paso adicional al algoritmo canónico, consiste en una estrategia de búsqueda *dimension learning-based hunting DLH*, donde cada lobo aprende de sus vecinos para generar un nuevo candidato a la posición con el valor más competitivo. En el *GWO* canónico, se rodea a la presa considerando la posición de los tres mejores lobos para generar el candidato para la nueva mejor posición denominada $X_{i-GWO}(t+1)$ (Ecuación 9).

La estrategia de búsqueda *DLH* se basa en el comportamiento de caza individual de los lobos, donde aprenden de la posición de sus vecinos, y cada dimensión de la nueva posición se calcula con la ecuación 9 seleccionando dos lobos al azar, uno de la población y otro del vecindario. Después, en lugar de $X_{i-GWO}(t+1)$, la estrategia de búsqueda *DLH* genera otro candidato para la posición más competitiva, llamado $X_{i-DLH}(t+1)$, Para esto, se calcula un radio $R_i(t)$ utilizando la distancia euclidiana entre la posición del lobo actual $X_i(t)$ y la posición del nuevo candidato $X_{i-GWO}(t+1)$, computado por la ecuación 18).

$$R_i(t) = \|X_i(t) - X_{i-GWO}(t+1)\| \quad (18)$$

Después, el vecindario del lobo actual $X_i(t)$ denotado por $N_i(t)$ es construido con la ecuación 19, tomando en cuenta el radio $R_i(t)$ calculado previamente.

$$N_i(t) = \{X_j(t) \mid D_i(X_i(t), X_j(t)) \leq R_i(t), X_j(t) \in \text{Pop}\} \quad (19)$$

Una vez que el vecindario de $X_i(t)$ es construido, se lleva a cabo el aprendizaje por la ecuación 20, donde la dimensión de $X_{i- DLH}(t+1)$, se calcula utilizando la dimensión de un vecino aleatorio $X_{n,d}(t)$ seleccionado de $N_i(t)$, y de un lobo aleatorio seleccionado de la población $X_{r,d}(t)$.

$$X_{i- DLH,d}(t+1) = X_{i,d}(t) + \text{rand} \times (X_{n,d}(t) - X_{r,d}(t)) \quad (20)$$

Etapas de selección y actualización: En esta etapa, $X_{i- GWO}(t+1)$ y $X_{i- DLH}(t+1)$ se comparan por su valor de aptitud (*fitness value*), con la ecuación 21.

$$X_i(t+1) = \begin{cases} X_{i- GWO}(t+1), & \text{if } f(X_{i- GWO}) < f(X_{i- DLH}) \\ X_{i- DLH}(t+1) & \text{otherwise} \end{cases} \quad (21)$$

Después, para actualizar la nueva posición, si el valor de aptitud de $X_i(t+1)$ es mejor que el valor de aptitud de $X_i(t)$, $X_i(t)$ se actualiza por el candidato seleccionado. En caso contrario, $X_i(t)$ permanece sin cambios en la población. Después de realizar este procedimiento para todos los individuos, el contador de las iteraciones se incrementa hasta que alcanza el número predefinido de iteraciones.

Algorithm 2: El Optimizador del Lobo Gris Mejorado (i-GWO)

Input: N , D , $Maxiter$ **Output:** Posible Óptimo Global

```
1 begin
2   Inicialización (Distribución aleatoria de  $N$  lobos en el espacio de
   búsqueda y calculo de su función de aptitud.
3 for  $iter = 2$  to  $Maxiter$  do
4   Encuentra  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ .
5   for  $i = 1$  to  $N$  do
6     Computar  $X_{i-GWO}(t + 1)$  con Eq.(9).
7     Calcular  $R_i(t)$  con Eq.(18).
8     Construcción del vecindario  $N_i(t)$  con el radio  $R_i(t)$ 
       calculado por Eq.(19).
9     for  $d = 1$  to  $D$  do
10       $X_{i-DLH,d}(t + 1) = X_{i,d}(t) + \text{rand} \times (X_{n,d}(t) - X_{r,d}(t))$ .
       Eq.(20).
11      Selección del mejor ( $X_{i-GWO}(t + 1)$ ,  $X_{i-DLH}(t + 1)$ ). Eq.(21).
12      Actualizar la población.
13 return El optimo global
```

2.4.3. Optimizador del Lobo Gris Binario *BGWO*

El Optimizador del Lobo Gris canónico está diseñado para resolver problemas continuos, donde los lobos cambian constantemente su posición hacia cualquier lado en el espacio de búsqueda. En la minería de reglas de asociación, las reglas están representadas como un vector binario dentro de la base de datos transaccional Figura 2. Debido a esta característica, la búsqueda de soluciones está restringida a un espacio discreto, donde las posiciones de cada lobo solo pueden tomar valores binarios $[0, 1]$ [33], de esta manera los lobos se mueven en las esquinas de un hipercubo, cambiando los bits de ceros y unos que determinan su posición, por lo que un lobo con 0 bits cambiados se puede decir que no se mueve, en cambio se puede decir que los que están constantemente cambiando sus bits se mueven con mayor frecuencia [34], Algoritmo 5.

En este enfoque los vectores de las posiciones de los lobos son forzados a ser binarios, donde el valor de cada posición es determinado probabilísticamente utilizando la ecuación 22, donde *rand* es un número aleatorio con distribución $[0, 1]$ y el argumento de la función sigmoide, ecuación 23, es el promedio de la posición de los tres mejores lobos.

$$x_d^{t+1} = \begin{cases} 1 & \text{if sigmoid} \left(\frac{x_1+x_2+x_3}{3} \right) \geq \text{rand} \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

$$\text{sigmoid}(a) = \frac{1}{1 + e^{-10(x-0,5)}} \quad (23)$$

Algorithm 3: El Optimizador del Lobo Gris Binario (BGWO)

Input: N, D, Maxiter

Output: Posible Óptimo Global

```
1 begin
2   ┌ Inicialización (Distribución aleatoria de N lobos en el espacio de
   └ búsqueda y cálculo de su función de aptitud.
3 for iter = 2 to Maxiter do
4   ┌ Encontrar  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ .
5   └ for  $i = 1$  to  $N$  do
6     ┌  $X_{i-GWO}(t + 1) = \begin{cases} 1 & \text{if sigmoid} \left( \frac{x_1+x_2+x_3}{3} \right) \geq \text{rand} \\ 0 & \text{otherwise} \end{cases}$ 
7     └ Selección del mejor ( $X_{i-GWO}(t + 1)$ ,  $X_i(t + 1)$ ).
   └ Actualizar la población.
8 return El optimo global
```

De igual manera, para la nueva posición generada por la técnica de búsqueda *DLH* (*Dimension learning-based hunting*) que implementa la versión mejorada del lobo gris, se utiliza la ecuación 22, para actualizar los vectores de posición probabilísticamente, donde solo se pueden tomar valores binarios $[0, 1]$, pero esta vez el argumento de la función sigmoide, es la ecuación 9, Algoritmo 6.

Algorithm 4: El Optimizador del Lobo Gris Binario Mejorado (i-BGWO)

Input: N, D, Maxiter

Output: Posible Óptimo Global

```
1 begin
2   Inicialización (Distribución aleatoria de N lobos en el espacio de
   búsqueda y cálculo de su función de aptitud.
3 for iter = 2 to Maxiter do
4   Find  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ .
5   for  $i = 1$  to  $N$  do
6      $X_{i-GWO}(t+1) = \begin{cases} 1 & \text{if sigmoid} \left( \frac{x_1+x_2+x_3}{3} \right) \geq \text{rand} \\ 0 & \text{otherwise} \end{cases}$ 
7     Calcular  $R_i(t)$  con Eq.(18)
8     Construcción del vecindario  $N_i(t)$  con el radio  $R_i(t)$ 
       calculado por Eq.(19.)
9     for  $d = 1$  to  $D$  do
10     $X_{i-DLH,d}(t+1) =$ 
        $\begin{cases} 1 & \text{if sigmoid} (X_{i,d}(t) + \text{rand} \times (X_{n,d}(t) - X_{r,d}(t))) \geq \text{rand} \\ 0 & \text{otherwise} \end{cases}$ 
11    Selección del mejor ( $X_{i-GWO}(t+1)$ ,  $X_{i-DLH}(t+1)$ ). Eq.(21).
12    Actualizar la población.
13 return El optimo global
```

3. Estado del Arte

La minería de reglas de la asociación *ARM* (Association Rule Mining por sus siglas en inglés) es una de las técnicas más importantes del aprendizaje máquina y de la minería de datos [35]. En la actualidad, este es un campo activo donde se desarrollan nuevos métodos e implementaciones para la generación y optimización de reglas de asociación, al igual para los valores paramétricos de las restricciones mínimas que involucran [4]. La computación evolutiva consiste en grupos de algoritmos de optimización que se inspiran en la evolución biológica, al tener un carácter estocástico y meta-heurístico pueden llegar encontrar soluciones cercanas a la más óptima para una gran cantidad de problemas de optimización [24]. El uso de algoritmos evolutivos para la minería de reglas de asociación ha demostrado ser potencialmente útil para el procesamiento de una gran cantidad de datos y descubrimiento de información relevante [12], unas características claves de este enfoque, es que permite el descubrimiento de reglas de asociación sin tener que realizar el paso de la generación de conjuntos de ítems frecuentes, y también generar de manera automática los valores apropiados para los umbrales de soporte y confianza mínimos, a partir de las mejores reglas generadas y poder utilizarlos en otros algoritmos para el minado de reglas de asociación, estos dos aspectos traen consigo una reducción considerable en el tiempo de ejecución en comparación con los enfoques tradicionales [4], siendo esta una de las razones por las que se le considera parte del estado del arte[4]. En las últimas dos décadas, se han presentado modelos para el descubrimiento de reglas de asociación basados en este tipo de algoritmos meta heurísticos (Figura 5). Este campo es conocido como Computación Evolutiva para la Minería de Reglas de Asociación (*Evolutionary ARM, EARM por sus siglas en ingles*). En la figura 5, se muestra la distribución anual del número de artículos científicos publicados referentes a esta área.

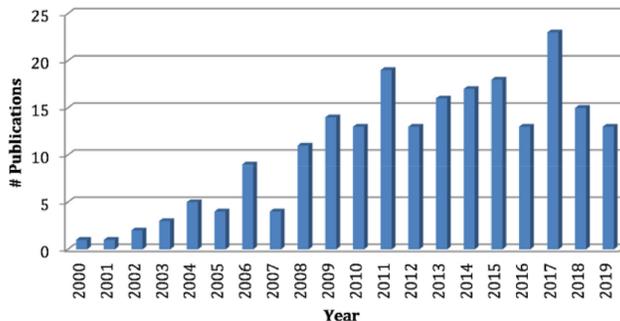


Figura 5: Distribución de las publicaciones relacionadas directamente a EARM, imagen tomada de [4].

Los dos algoritmos tradicionales más utilizados para minar reglas de asociación en la literatura son, FP-Growth [11] y Apriori [8], el último siendo pionero en la extracción de conjuntos de datos frecuentes de una base de datos transaccional. Estos dos algoritmos forman la base de donde parten muchos otros, los cuales implementan mejoras considerables en su rendimiento [10], aunque aún requieren una cantidad considerable de cómputo, memoria y tiempo ejecución. La complejidad de los algoritmos basados en *Apriori* son afectados directamente por el número de transacciones e ítems existentes en la base de datos, lo que aumentan el tiempo de ejecución exponencialmente, esto se debe a que el algoritmo recorre la base de datos varias veces para calcular la frecuencia de aparición de los diferentes conjuntos de ítems, convirtiéndose en un problema NP-Hard [4].

Debido a esta complejidad, es necesario utilizar parámetros para limitar el espacio de búsqueda. El más fundamental es la frecuencia mínima de ocurrencia de una regla de asociación, la cual lleva el nombre de soporte. Utilizar la frecuencia de ocurrencia de las asociaciones para disminuir la cantidad de reglas que se generan no es práctico, debido a que las reglas con una frecuencia de aparición cerca del 100 % son habitualmente conocidas, llegando a ser triviales y por lo tanto sin proporcionar un conocimiento valioso. Las reglas que realmente interesan suelen aparecer con frecuencias menores. En muchos casos se define un umbral muy pequeño para el parámetro de la frecuencia, por lo que los modelos arrojan demasiadas reglas de asociación, las cuales son difíciles de analizar por un agente humano, a este problema en la literatura científica [36, 37] se le conoce como el problema de reglas redundantes debido a que varias reglas describen las mismas filas de la base

de datos, este es uno de los principales inconvenientes para la utilización de modelos basados en reglas de asociación. Actualmente el desarrollo de algoritmos eficientes para el minado de reglas de asociación es un campo activo, en el que se siguen desarrollando investigaciones [10, 4].

En la mayoría de las investigaciones sobre Computación Evolutiva para Reglas de Asociación solo se han utilizado los algoritmos más populares del área. Una versión mejorada del PSO [38], fue el primer acercamiento de este modelo a la minería de reglas de asociación. En [39] se describió formalmente y por primera vez el uso de algoritmos genéticos GA aplicados a las reglas de asociación. La evolución diferencial multi-objetivo basado en principio de Pareto [40], fue la primera contribución del modelo en el área de la Computación Evolutiva para Reglas de Asociación. En el siguiente diagrama se muestra la taxonomía y distribución de los algoritmos evolutivos utilizados en las publicaciones sobre *EARM*. Se puede observar una clara inclinación hacia la utilización de los modelos basados en algoritmos genéticos, representando el 55 % del total, seguido por los algoritmos basados en inteligencia de enjambre con el 37.4 % y el resto de modelos representando el 7.6 %.

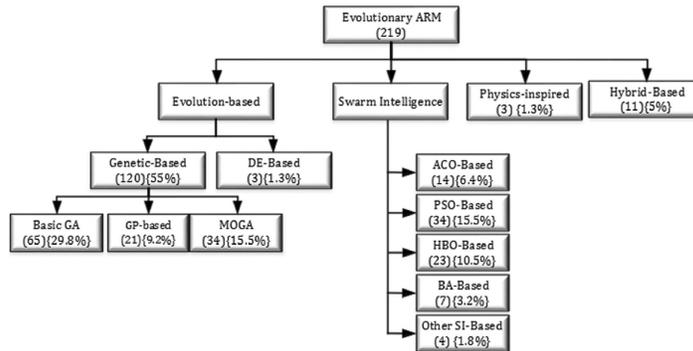


Figura 6: Taxonomía de los algoritmos evolutivos usados en la minería evolutiva de reglas de asociación EARM, imagen tomada de [4].

Los enfoques de Computación Evolutiva para Reglas de Asociación, abarcan una gran variedad de dominios. En general, las principales aplicaciones se pueden agrupar en ocho categorías: Análisis de canasta de compra, sistemas de recomendación, redes informáticas, sanidad, medio ambiente, industria, educación y tráfico vehicular [4]. A continuación, se muestra un diagrama donde se representa el porcentaje de la distribución de dominios en donde se utiliza la *EARM* acorde a los artículos publicados. Se puede observar que

el análisis de canasta de mercado representa el 11 % del total, demostrando ser un área relevante para la comunidad y con una gran oportunidad de investigación. Trabajos recientes han realizado comparaciones de los algoritmos tradicionales para el minado de reglas de asociación con una aplicación enfocada al análisis de canasta de mercado en [41, 42, 43].

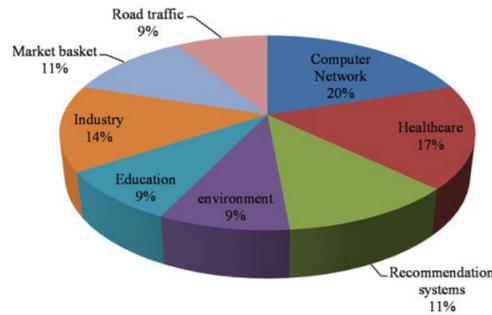


Figura 7: Distribución de los diferentes dominios donde se utiliza la minería evolutiva de reglas de asociación EARM[4].

Dentro de la computación evolutiva, el optimizador del lobo gris *GWO* es un algoritmo que imita la jerarquía de liderazgo y el mecanismo de caza de los lobos grises en la naturaleza, este forma parte de los modelos basados en inteligencia de enjambre (*Swarm Intelligence SI* por sus siglas en inglés) [44]. En 2020, se realizó una revisión semántica de la literatura sobre la minería evolutiva de reglas de asociación [4] (figura 6), donde se menciona que los algoritmos de inteligencia de enjambres *SI-based*, están presentes en el 37.4 % de las publicaciones del área. El optimizador del lobo gris *GWO* se encuentra dentro de la categoría de *Other SI-based*, el cual representa el 1.8 % de toda la investigación, haciéndolo un candidato perfecto para la exploración de nuevas oportunidades de investigación en el área de la Computación Evolutiva para Reglas de Asociación.

En [45] se menciona que, aunque la minería de reglas de asociación es una área popular en la actualidad, la utilización de métodos bio-inspirados para esta tarea sigue perteneciendo a una minoría, por esta razón existe una gran oportunidad de investigación en el área, aunque una desventaja es que hay una notable deficiencia en los análisis teóricos sobre los algoritmos basados en inteligencia de enjambres para minar reglas de asociación. También se menciona que proveer de códigos fuentes al igual que cursos sobre el tema

ayudaría a esparcir estos enfoques en aplicaciones hacia datos del mundo real.

El optimizador del lobo gris *GWO* se ha utilizado para resolver una gran diversidad de problemas en varios dominios [46]. En la investigación biomédica, se propuso un método híbrido que utilizó *GWO* para optimizar redes *MLP* (Multilayer Perceptron por sus siglas en inglés) y de esta manera poder detectar melanomas malignos a partir de imágenes [47]. En el ámbito de la ingeniería, se resolvió un problema de planificación de ruta para un vehículo aéreo, donde la metodología propuesta consideró los problemas y limitaciones del campo de batalla, logrando obtener caminos óptimos [48]. En [49] se utilizó *GWO* para la tarea de modificar una base de datos de tal manera que ciertas reglas de asociación sensibles desaparezcan, sin afectar a los demás reglas no sensibles. En [50] se utilizó *GWO* para el minado de *conjuntos de datos* de alta utilidad *High Utility Itemsets HUI* y que tengan una importancia significativa, el algoritmo se usó para minar *HUI* utilizando cinco diferentes operadores booleanos. En [51] se utilizó el optimizador del lobo gris para la tarea de minería de reglas de clasificación cuantitativas con un enfoque multi objetivo. Además de los trabajos mencionados, son casi inexistentes otros trabajos donde se haya utilizado el *GWO* para minar reglas de asociación. La utilización de este algoritmo para la generación de reglas de asociación ha sido poco explorado en investigaciones previas del área, realizar este trabajo de investigación puede abrir nuevas oportunidades de investigación e innovación en el área de la computación evolutiva para la minería de reglas de asociación.

El optimizador mejorado del lobo gris también ha demostrado ser eficaz en otros dominios, Un ejemplo de esto son los resultados obtenidos en el artículo donde fue presentado [14], en este se puso a prueba el *i-GWO* contra otros algoritmos del área evolutiva para optimizar las variables de diseño de problemas de optimización en ingeniería, donde *i-GWO* demostró llegar a mejores resultados en la mayoría de evaluaciones.

3.1. Funciones de aptitud propuestas en la literatura.

La Función de Aptitud o *Fitness Function* es el valor que decide qué tan bueno es un individuo de la población en los algoritmos bio-inspirados, dependiendo del dominio del problema cada función de aptitud puede tomar características específicas para evaluar a la posible solución. En el caso del cómputo bio-inspirado para la minería de reglas de asociación, se han propuesto varias en la literatura, aunque estas consisten en estructuras dife-

rentes, todas coinciden en que utilizan las métricas básicas de las reglas de asociación (Soporte y Confianza) en sus fórmulas.

Una de las primeras fue propuesta en [52], donde se utilizó un algoritmo híbrido (PSO/ACO), el cual no requiere de las especificaciones de los umbrales de soporte y confianza, debido a que en cada iteración se seleccionan las mejores reglas generadas según los valores de la función de aptitud:

$$Fitness (X \Rightarrow Y) = (support (X \Rightarrow Y) - (marked * w) - (amplitude * \psi) + (n Atr * \mu) \quad (24)$$

Donde el soporte ($X \Rightarrow Y$) representa el soporte de la regla, *marked* indica que un registro ha sido cubierto previamente por un *itemset*. Para penalizar los registros cubiertos, se utiliza el factor (*w*). *Amplitude* penaliza la amplitud de los intervalos que conforman el *itemset*. El factor ψ permite el crecimiento de los intervalos y el parámetro número de atributos (*nAtr*) recompensa los *conjuntos de datos* frecuentes con un mayor número de ítems, valor que incrementa o disminuye por medio del factor μ .

En [26], se propone un algoritmo basado en la versión binaria de PSO para minar reglas de asociación de bases de datos de transaccionales. La metodología de este trabajo de investigación consiste en dos partes, preprocesamiento y minería. En la primera etapa, los datos son transformados en formato binario. En la segunda etapa, se utiliza el PSO binario para minar las reglas de asociación. En este trabajo se propuso utilizar el producto del soporte y confianza como función de aptitud:

$$Fitness (X \Rightarrow Y) = Supp(X \Rightarrow Y) * Con f(X \Rightarrow Y) \quad (25)$$

Esta función de aptitud tiene la ventaja de que cuando se multiplican los valores del soporte y confianza, los *conjuntos de datos* que tienen valores de soporte bajos, pero generan reglas interesantes no son eliminados. En la segunda etapa de esta metodología, se selecciona únicamente la mejor regla encontrada por cada ejecución. Este proceso es repetido M veces hasta que las M reglas son encontradas, haciendo M un parámetro del algoritmo, para la experimentación se utilizaron dos conjuntos de datos, Charles Book Club y uno referente a una tienda de conveniencia.

En [5], también se propuso un algoritmo basado en la versión binaria de PSO para minar reglas de asociación de bases de datos de transaccionales,

en este algoritmo son determinados automáticamente los valores de los umbrales de soporte y confianza. El algoritmo consiste también en dos etapas, preprocesamiento y minería. En la primera etapa, los datos se transforman a formato binario y en la segunda etapa se minan las reglas de asociación. En cada iteración, la aptitud de cada partícula se calcula acorde a su soporte, confianza y longitud del número de ítems presentes en la regla:

$$\text{Fitness}(X \Rightarrow Y) = \text{Conf}(X \Rightarrow Y) * \log(\text{Supp}(X \Rightarrow Y) * \text{Length}(X \Rightarrow Y) + 1) \quad (26)$$

Una vez terminado el procedimiento de búsqueda por el PSO binario (hasta que se encuentran las M reglas, definidas por el parámetro M), se puede recomendar el promedio de los valores obtenidos del soporte y confianza de las mejores reglas encontradas, como umbrales para la base de datos seleccionada, estos pueden ser empleados por otros algoritmos para el minado de reglas de asociación. La experimentación se realizó con la base de datos *Foodmart 2000*.

4. Metodología

Como se mencionó en la sección 2.1.1, para llevar a cabo este proyecto se utilizó la metodología CRISP-DM, la cual consiste en 6 etapas. Por fines prácticos en esta investigación, solo se llegará hasta la etapa 5, debido a que la sexta etapa queda fuera del enfoque de esta investigación:

4.1. Etapa 1: Entendimiento del problema.

En esta trabajo de investigación se propone la implementación del optimizador del lobo gris mejorado para la generación de reglas de asociación, de esta manera, encontrar las posibles asociaciones que existen entre ítems pertenecientes a bases de datos transaccionales. El objetivo principal consiste en generar reglas de asociación de mayor o igual calidad en comparación con otras propuestas existentes en el estado del arte.

El algoritmo propuesto *i-BGWO* consta de 2 fases: preprocesamiento y minado Figura (12). A continuación se exponen detalladamente las tareas que se llevaron a cabo en cada etapa siguiendo la metodología CRISP-DM.

4.2. Etapa 2: Entendimiento de los datos.

Para esta investigación se utilizaron siete bases de datos (Cuadro 2), provenientes de distintas fuentes, la mayoría de estos conjuntos de datos fueron tomados debido a que se han utilizado en tres de los trabajos más relevantes de la literatura en el área evolutiva para la minería de reglas de asociación [5] [26] [53] y sirven como punto de comparación para el trabajo actual.

Bases de Datos				
Nombre	Tamaño	Filas	Columnas	Origen
Charles Book Club	82.1 Kb	4001	10	Github [54]
MovieLens	2.36 Mb	100,837	4	Kaggle [55]
Foodmart 2000	4.31 Mb	86,837	6	Microsoft SQL Server [56]
Wine	10.5 Kb	178	14	UCI Machine Learning Repository [57]
Zoo	4.02 Kb	101	18	UCI Machine Learning Repository [57]
Wisconsin Breast Cancer	19.4 Kb	699	11	UCI Machine Learning Repository [57]
Mi Tiendita	71.61 Mb	912,102	12	[Privado]

Tabla 2: Conjuntos de datos utilizados.

Tres bases de datos fueron tomadas de *UCI Machine Learning Repository*, el conjunto de datos *MovieLens* fue tomado del repositorio de Kaggle, mientras que *Charles Book Club* se consiguió a través de la plataforma

GitHub, *Foodmart 2000* fue extraída específicamente de Microsoft SQL Server 2000, por último y para aplicar el algoritmo a un caso de estudio real se proporcionó por parte de una tienda de conveniencia, localizada en Xalpa, una base de datos correspondiente a las transacciones realizadas en un punto de venta *POS Point of Sales por sus siglas en inglés*, donde se recolectaron las ventas realizadas en el periodo de enero de 2016 a marzo de 2017. Cabe destacar que ésta se utilizó previamente en [58], donde se propusieron técnicas del aprendizaje supervisado para identificar y pronosticar la ausencia de venta de productos (venta nula). Todos los conjuntos de datos tuvieron que ser preprocesados para darles una naturaleza discreta y de esta manera poder utilizarlos en el algoritmo propuesto.

4.3. Etapa 3: Preparación de los datos.

Esta etapa consiste en realizar el preprocesado de los datos, al igual que su preparación de estos para su utilización en el algoritmo propuesto. La mayoría de las bases de datos contaron con diferentes procesos de transformación y modelado para poder ser discretizadas, en general se realizaron las siguientes tareas para poder transformarlas:

- Para las bases de datos que no cuentan con el formato transaccional horizontal expuesto en la sección 2.3, se realizan agrupaciones acorde a los números de identificación correspondiente para cada transacción.
- Con la base de datos en formato transaccional, donde cada fila corresponde a una transacción, esta se transforma al formato booleano, con el objetivo de eficientar el cálculo del soporte y la confianza, con las formulas expuestas en la sección 2.2.1. Es importante mencionar que la cuenta de la frecuencia de cada ítem se calcula de manera vertical, trasponiendo la base de datos (Figura 8).
- Se eliminan los ítems que no sobrepasan el umbral de soporte, para esto se realiza un análisis exploratorio y dependiendo del tamaño o dominio de la base de datos se asigna el umbral de soporte mínimo.
- Se eliminan las transacciones que contienen menos de 2 ítems, de esta manera se excluyen del proceso las transacciones que no generan ninguna regla de asociación, debido a que solo se pueden generar reglas a partir de 2 ítems como se mencionó en la sección 2.2, aunque este valor puede variar si se busca realizar una exploración más específica en un rango arbitrario de ítems por transacción.

- Se devuelve un objeto, el cual contiene la base de datos limpia y una lista con los soportes de cada ítem en el orden correspondiente a la base de datos.

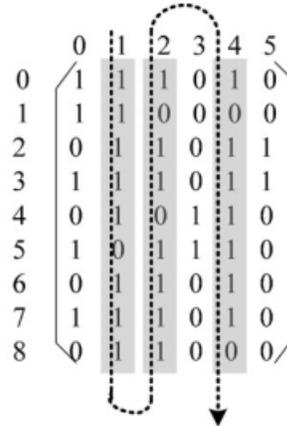


Figura 8: Método de búsqueda vertical para contar la frecuencia de cada ítem [5].

4.3.1. Base de Datos Mi Tiendita

Con el objetivo de descubrir reglas de asociación entre los productos que se compran juntos en una tienda de conveniencia ubicada en Xalapa, Veracruz, fue proporcionada una base de datos proveniente de un punto de venta (Point of Sales POS, por sus siglas en inglés) del negocio. La base de datos fue suministrada por el departamento de sistemas, el cual funge como responsable de todos los procesos tecnológico administrativo de la empresa. Se proporcionaron tres archivos en formato TXT (ver Tabla 3), recolectados en el periodo de enero de 2016 a marzo de 2017, cabe destacar que los nombres de los ítems en la base de datos están representados por su código de artículo (Stock Keeping Unit SKU, por sus siglas en inglés).

Nombre del Archivo	Tamaño en Megabytes
Archivo 1	26 Mb
Archivo 2	27Mb
Archivo 3	18 Mb

Tabla 3: Archivos origen de base de datos (Mi Tiendita).

Estos archivos contienen todas las transacciones correspondientes al periodo del primero enero de 2016 al trece de marzo de 2017. En las tablas 3 a la 7 se presenta la estructura de los datos:

Variable	Descripción
Sucursal	Número de la sucursal
Fecha	Fecha y hora de la venta
Caja	Número de caja donde se realizó la venta
Número de venta	Identificador de cada venta
Clave del producto	Identificador de cada producto
Cantidad de producto	Cantidad vendida de un producto en una venta
Precio	Precio del producto para el cliente
Iva	Impuesto al valor agregado
Usuario	Clave del empleado que atendió
Corte de caja	Cierre de caja perteneciente a una venta
Corte de caja parcial	Cierre de caja parcial perteneciente a una venta
Costo	Costo del producto por parte del proveedor

Tabla 4: Descripción general de los atributos.

Variable	Tipo	Densidad	Escala de medición
Sucursal	Cualitativa		Ordinal
Fecha	Cuantitativa	Continua	
Caja	Cualitativa		Ordinal
No venta	Cualitativa		Ordinal
Clave product	Cualitativa		Nominal
Cant producto	Cuantitativa	Continua	Relación
Precio	Cuantitativa	Continua	Relación
Iva	Cuantitativa	Continua	Relación
Usuario	Cualitativa		Nominal
Corte	Cualitativa		Ordinal
Corte Parcial	Cualitativa		Ordinal
Costo	Cuantitativa	Continua	Relación

Tabla 5: Descripción de tipo de atributos.

Variable	Descripción
Sucursal	Número de la sucursal
Fecha	Fecha y hora de la venta
Caja	Número de caja donde se realizó la venta
Número de venta	Identificador de cada venta
Clave del producto	Identificador de cada producto
Cantidad de producto	Cantidad vendida de un producto en una venta
Precio	Precio del producto para el cliente
Iva	Impuesto al valor agregado
Usuario	Clave del empleado que atendió
Corte de caja	Cierre de caja perteneciente a una venta
Corte de caja parcial	Cierre de caja parcial perteneciente a una venta
Costo	Costo del producto por parte del proveedor

Tabla 6: Descripción general de los atributos.

Variable	Característica
Fecha	Primer registro: 2016-01-01 00:01:18
	Último registro: 2017-03-13 09:03:42
	Se registraron ventas 438 días
No. de venta	Es un número secuencial que incrementa una entrada por cada artículo de una transacción.
	Hay un total de 556,921 transacciones distintas
Clave del producto	Existen 3,491 productos distintos
	Valor con mayor frecuencia: 75007614 con 19,115 entradas.
Cantidad de producto	Recuento: 912,102.00
	Suma: 1,098,831.95
	Media: 1.204
	Mín: -96.00
	Max: 453.29
	Std: 1.65
Precio	Recuento: 912,102.00
	Suma: 16,814,996.063
	Media: 18.43
	Mínimo: 0.00
	Máximo: 1266.00
	Std: 22.003
Costo	Recuento: 912,102.00
	Suma: 12,464,895.083
	Media: 13.66
	Mínimo: 0.00
	Máximo: 890.80
	Std: 16.95

Tabla 7: Estadísticas descriptivas de la base de datos.

Al realizar una exploración, se hallaron varias anomalías e inconsistencias que dificultan el análisis y procesado de la base de datos, por lo que fue

limpiada para descartar datos y atributos que solo causarían ruido en fases posteriores, siendo las más relevantes:

- Descartar atributos que no aportan valor al objetivo de la investigación (Sucursal, Caja, Tasa iva, Corte, y Corte parcial).
- Solo se tomó en cuenta el periodo de 2016 debido a datos faltantes en el año de 2017.
- Se eliminaron todas las entradas con el valor "SOBRANTE" del atributo CLAVE, debido a que este corresponde al cierre de caja, el cual no genera ningún interés en la motivación de la investigación.
- Se eliminaron los valores iguales o menores a cero en las columnas: Cantidad de producto, Precio y Costo, debido a que estos registros podrían representar errores del sistema o devoluciones, los cuales quedan fuera del enfoque de proyecto.

No. De productos en una transacción	Recuento de transacciones
1	313830
2	108288
3	41190
4	16613
5	7326
6	3617
7	1817
8	935
9	493
10	302
11	171
12	105
13	65
14	44
15	23
16	14
18	10
17	5
19	5

Tabla 8: Cantidad de productos por transacción.

La tabla 8 muestra el número de ítems vendidos por la cantidad de transacciones existentes, se puede observar que las transacciones más frecuentes son las que contienen menos ítems, siendo irrelevantes las que solo contienen uno para enfoque de esta investigación, debido a que, como se mencionó en la sección 2.2, para poder generar una regla de asociación se necesitan por lo menos dos ítems.

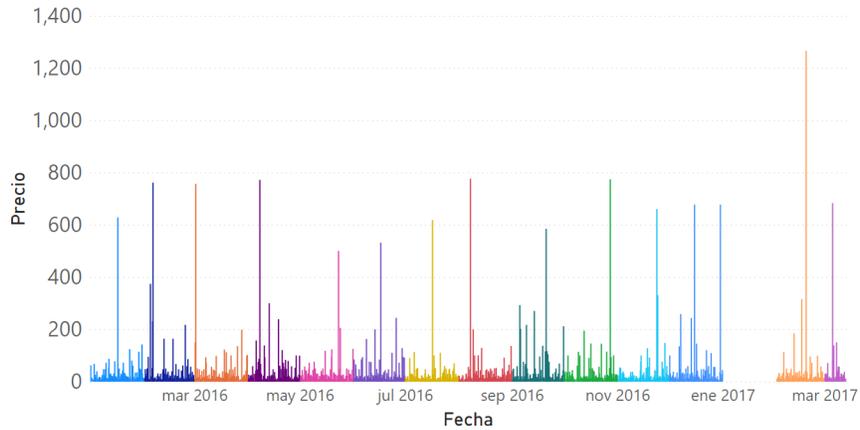


Figura 9: Serie de tiempo con el costo de cada transacción por mes.

En la figura 9 se presenta el costo total de cada venta en la serie de tiempo, mostrando que se obtiene una consistencia en la distribución para cada periodo.

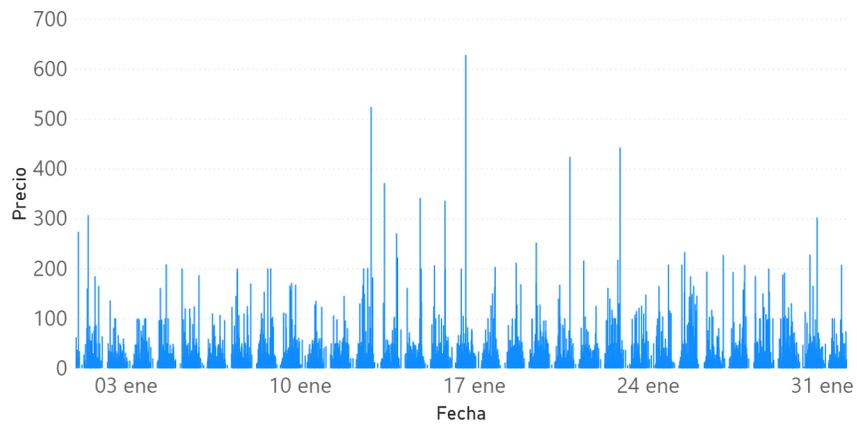


Figura 10: Ventas de enero 2016.

La figura 10 muestra solo las transacciones del mes de enero, a simple vista se pueden distinguir los surcos que existen entre los periodos diurno y nocturno del día, donde es notorio que las ventas disminuyen durante la noche. Las ventas totales en el mes de enero son 1.17 millones de pesos mexicanos.

4.3.2. Representación de las bases de datos transaccionales

En esta propuesta, la base de datos transaccional se representa en formato binario a través de vectores horizontales, donde 1 denota la presencia del ítem y 0 denota la ausencia del mismo. En la Tabla 10 se muestra la representación binaria del conjunto de datos transaccional. 9

<i>T1</i>	a,b,d,e,f
<i>T2</i>	b,c,e
<i>T3</i>	a,b,d,e
<i>T4</i>	a,b,c,e
<i>T5</i>	a,b,c,d,e,f
<i>T6</i>	b,c,d

Tabla 9: Base de datos transaccional.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>T1</i>	1	1	0	1	1	1
<i>T2</i>	0	1	1	0	1	0
<i>T3</i>	1	1	0	1	1	0
<i>T4</i>	1	1	1	0	1	0
<i>T5</i>	1	1	1	1	1	1
<i>T6</i>	0	1	1	1	0	0

Tabla 10: Base de datos transaccional binarizada.

Como un ítem no frecuente no puede formar parte de un *itemset* frecuente debido al principio de la propiedad de clausura descendiente de soporte sección 2.3.2, todos los ítems que no cumplen con el umbral de mínimo soporte son excluidos de la base de datos binarizada.

4.4. Etapa 4: Modelado predictivo.

Esta etapa lleva a cabo la minería de reglas de asociación, siendo la segunda parte del algoritmo propuesto, esta consta de los siguientes pasos (Figura 12):

- Se codifica la base de datos binaria, seleccionando aleatoriamente al consecuente.
- Se selecciona a la manada aleatoriamente.
- Se calcula el soporte, confianza y aptitud de los individuos seleccionados.
- El Optimizador del Lobo Gris Binario Mejorado *i-BGWO* empieza la búsqueda de la mejor solución (regla de asociación) y se selecciona a la mejor (Alfa).
- Mientras el mejor individuo no se encuentre en la lista con las mejores reglas encontradas y mientras que no se hayan conseguido las M reglas predefinidas en el parámetro de entrada, se continua buscando reglas hasta cumplir con las condiciones.

4.4.1. Representación de las reglas de asociación.

Para representar las reglas de asociación en una base de datos transaccional binaria, la literatura ofrece dos formas en las que pueden ser codificadas: el enfoque Pitsburg [59], donde cada individuo es representado por varias reglas de asociación y el Enfoque Michigan[59], en el cual, cada solución representa una regla (antecedente y consecuente). En esta investigación, se utiliza el enfoque Michigan como en [26], debido a que este representa una regla en un solo vector o individuo.

Para codificar una regla con el enfoque Michigan, se define un vector V_n de tamaño n , donde n es la cantidad de ítems presentes en la base de datos, cada ítem puede tomar tres estados:

- $V_i = 11$, se refiere a la que el ítem en la posición i pertenece al consecuente de la regla.
- $V_i = 10$, se refiere a la que el ítem en la posición i pertenece al antecedente de la regla.
- $V_i = 00$, se refiere a la que el ítem en la posición i no está incluido en la regla.

Por ejemplo, seleccionando la primera transacción del Tabla 9 $T1 = a, b, d, e$ y si se quiere representar la regla $R1 = [c, b] \Rightarrow [e]$ esta se representaría de la siguiente manera: $R1 = [00,10,10,11]$

4.4.2. Función de aptitud *Fitness*.

Para calificar la calidad de las reglas obtenidas con *i-BGWO*, se emplea la función propuesta en [5], a cual involucra el soporte, confianza y cantidad de ítems el *itemset*, (Ecuación 27).

$$\text{Fitness}(X \Rightarrow Y) = \text{Conf}(X \Rightarrow Y) * \log(\text{Supp}(X \Rightarrow Y) * \text{Length}(X \Rightarrow Y) + 1) \quad (27)$$

En esta ecuación los valores de soporte y longitud del *itemset* son multiplicados para excluir las reglas cuyos soportes son iguales a cero. Se suma 1 a la multiplicación de longitud con el soporte para evitar que la función de logaritmo base 10 de como resultado cero y de esta manera darle un peso similar con respecto al valor de la confianza. El objetivo de esta función de *fitness* es la maximización, por lo que, mientras más grande sea el resultado mayor será la aptitud del individuo. Esta ecuación permite que los ítems menos frecuentes en la base de datos pero que pueden llegar a ser potencialmente importantes, no se excluyan y se prioricen las reglas que cuentan con un mayor número de ítems.

4.4.3. Algoritmo para el minar reglas de asociación *i-GBWO*.

El Optimizador del Lobo Gris Binario Mejorado *i-BGWO* utiliza los siguientes parámetros de entrada: las reglas que se desean generar *M Reglas*, el conjunto de datos después de ser preprocesado, el número de iteraciones que tomará el *i-BGWO* para generar las reglas de asociación y el tamaño de la manada (población).

En *i-BGWO* cada vector contiene la posición de un lobo (individuo) candidato posible a la mejor solución $V_i = (i_1, i_2, \dots, i_n)$. Los vectores están constituidos por valores binarios $[0, 1]$, aunque estos también pueden tomar los valores de $['11', '10', '00']$ sección 4.4.1. Los vectores toman el tamaño del número de columnas *ítems* que hay en la base de datos. La mejor solución se denomina alfa α , seguida por la segunda Beta β y por último Delta δ , el resto de las soluciones candidatas son consideradas Omegas ω . El proceso de búsqueda de una solución altamente competitiva es guiada por los tres mejores, acorde al algoritmo descrito en la sección 2.5.1.

En la Figura 12, se describen los pasos del algoritmo propuesto para minar reglas de asociación: Después procesar la base de datos, se transforma en formato binario, posteriormente se codifican las transacciones en reglas de asociación arbitrarias, donde se elige al consecuente de manera aleatoria utilizando el enfoque Michigan sección 4.4.1, seguido se selecciona una manada de lobos (población de individuos) aleatoriamente y sin repetición, acto siguiente, se calcula el soporte y la confianza de cada lobo en la manada sección 2.2.1, con estos valores se calcula la aptitud de cada lobo en la manada con la ecuación 27. Con la aptitud generada, empieza el proceso de búsqueda con *i-BGWO* sección 2.5.3, para la tarea de minar reglas de asociación, se necesita tomar en consideración dos aspectos importantes, la codificación y decodificación del vector de posición del lobo, de binario al formato generado por el enfoque Michigan o al contrario y sin volver a seleccionar aleatoriamente al antecedente y al consecuente, esto se realiza para poder calcular las nuevas posiciones de X_{i-GWO} y X_{i-DLH} , a través de las formulas expuestas en la sección 2.5.2. También se utiliza para calcular las métricas de soporte, confianza y longitud de la regla, las cuales son necesarias para poder determinar la aptitud de la nueva posición generada.

El optimizador continúa cambiando los valores de cada dimensión de los vectores de posición hasta que se llegue a la condición de paro predefinida en los parámetros de entrada, también tomando en consideración solo las dimensiones *ith* de los vectores que denotan la presencia de un ítem, de esta manera los valores se mueven en las esquinas de un hipercubo, cambiando el estado de sus bits *ítems* de 10 a 11, los cuales determinan su posición y aptitud. Una vez llegada la condición de paro, se verifica si la mejor regla de la iteración actual ya ha sido agregada a la lista de reglas M , si la regla ya se encuentra en la lista, se repite el proceso hasta que se genere una regla que no había sido encontrada previamente, por lo que, mientras no se completen las M reglas introducidas en el parámetro de entrada, *i-BGWO* seguirá buscando hasta encontrar el número de reglas deseadas.

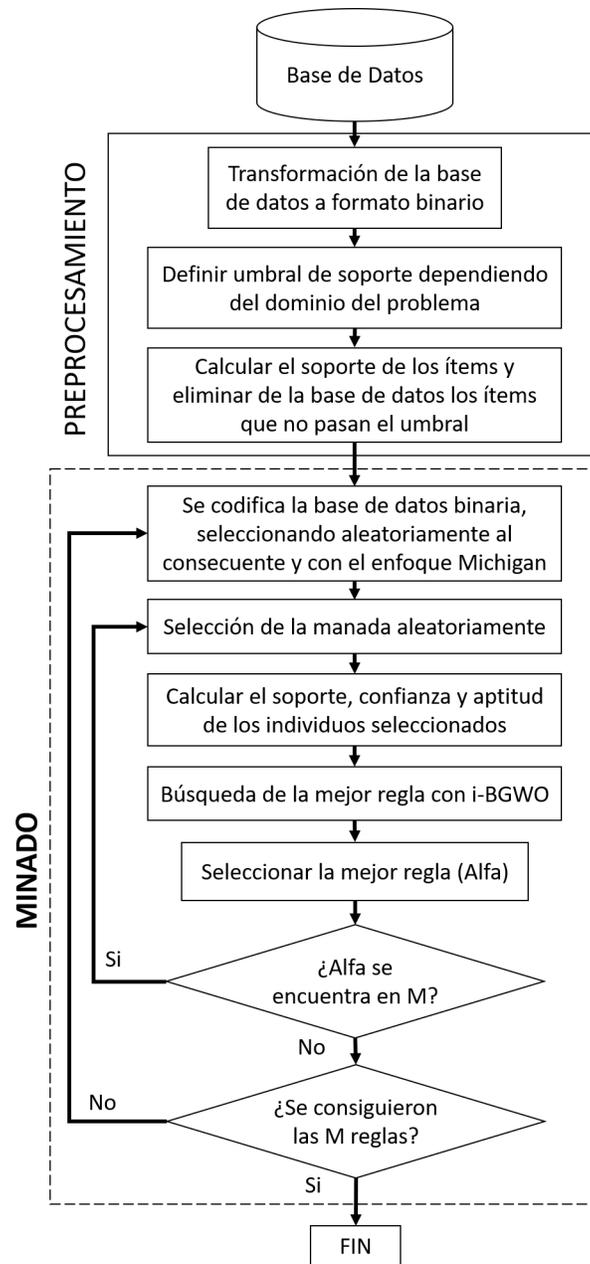


Figura 12: Metodología implementada para la programación del algoritmo *i-BGWO*.

4.5. Etapa 5: Evaluación.

El algoritmo propuesto fue comparado con los resultados obtenidos en [26], [5], [53] con el objetivo de comprobar su rendimiento, en estos artículos se utilizó la versión binaria del PSO para minar reglas de asociación, también el *i-GBWO* fue comparado con el Optimizador del Lobo Gris canónico y de esta manera confrontar los algoritmos acorde a la calidad y tiempo de ejecución de las reglas generadas. En la siguiente sección se especifica la experimentación realizada.

5. Resultados

Con el objetivo de evaluar el rendimiento que obtuvo el algoritmo propuesto para minar reglas de asociación, se comparó la calidad de las reglas obtenidas con otros algoritmos del estado del arte, al igual que el tiempo de ejecución en comparación con otros dos algoritmos implementados para este trabajo (*BGWO* y *BPSO*). Para validar la calidad de las reglas generadas con el modelo propuesto, se tomó el promedio de la función de aptitud y la confianza de las mejores reglas generadas, como se realizó en tres de los trabajos más relevantes de la literatura en el área [5], [26], [53], en cuanto al tiempo de ejecución, se consideró la cantidad de ítems y transacciones para comparar el *i-BGWO* contra las implementaciones del *BGWO* canónico, *BPSO*. También se agregaron las estadísticas descriptivas de los resultados, al igual que la prueba no paramétrica Wilcoxon-Mann-Whitney *WMW* para evaluar si existe una diferencia significativa entre los valores de aptitud y confianza generados por *i-BGWO* contra los otros algoritmos implementados en este trabajo.

5.1. Especificaciones técnicas de las pruebas realizadas.

Las pruebas de los algoritmos descritos previamente se realizaron en el sistema operativo Windows 10, con un procesador Intel Core i7 y 12 Gb de memoria RAM. Los algoritmos implementados (*i-BGWO*, *BGWO* y *BPSO*) fueron programados en el lenguaje Python 3.9, en cuanto al algoritmo *Apriori* se utilizó el módulo *Apyori*. Para las pruebas, se utilizaron siete bases de datos descritas en la sección 4.2.

Debido a que el *i-BGWO* no había sido utilizado para minar reglas de asociación previamente en la literatura, se optó por utilizar los parámetros de entrada con los que se obtuvieron mejores resultados para cada base de datos, después de ir iterando heurísticamente los valores de las variables. Para la implementación de *BPSO* se tomaron los parámetros utilizados por los autores de los trabajos con los que se están comparando los resultados de esta investigación: $w = 1$, $c1 = 2$ $c2 = 2$. Para los tres algoritmos implementados, se escogió una población de 10 individuos con 10 generaciones. Los algoritmos se ejecutaron 15 veces, de donde se tomó la ejecución con mejor rendimiento. Para *Apriori* se utilizaron como umbrales de soporte y confianza mínima los valores de las mejores reglas obtenidas con *i-BGWO*.

5.2. Comparación de la calidad de las reglas generadas.

Para determinar la calidad de las mejores reglas generadas y como se ha expuesto en otros trabajos del área, se calcula el promedio de la aptitud y confianza en el caso de *Apriori*. Mientras mayor sea el promedio obtenido, mayor será la calidad de la solución propuesta, esto se debe a la condición de maximización de la función de aptitud elegida (ecuación 27). En la Tabla 11, se muestran los mejores promedios de aptitud obtenidos en esta investigación y en los trabajos con los que se compara. Se puede apreciar que el modelo propuesto *i-BGWO*, obtuvo mejores resultados en la mayoría de las pruebas realizadas para el promedio de la función de aptitud de las 10 mejores reglas generadas. Cabe resaltar que los campos vacíos de las tablas 11 y 13 representados por un guión (-), denotan la falta de utilización de la base de datos en los trabajos desplegados en los títulos de las columnas.

Base de datos	BPSO	BGWO	i-BGWO	BPSO [26]	PSO-GES [53]
Charles Book Club	0.34318	0.33261	0.34484	0.08	0.11
MovieLens	0.25566	0.20274	0.29386	-	-
Foodmart 2000	0.06671	0.06422	0.07066	-	-
Wine	0.42883	0.46231	0.47134	-	0.22
Zoo	0.64157	0.60371	0.61276	-	0.62
Wisconsin Breast Cancer	0.71310	0.69984	0.71996	0.379	0.94

Tabla 11: Valor de aptitud promedio de las reglas obtenidas por los algoritmos.

BASE DE DATOS	ESTAD.	BPSO	BGWO	I-BGWO
Charles Book Club	mejor	0.47633	0.47633	0.47633
	promedio	0.34318	0.33261	0.34484
	dsv. est.	0.09542	0.10588	0.08516
	peor	0.16885	0.14527	0.2418
MovieLens	mejor	0.84367	0.73546	0.84367
	promedio	0.25566	0.20274	0.29386
	dsv. est.	0.29338	0.19887	0.27798
	peor	0.01175	0.05467	0.01255
Foodmart 2000	mejor	0.12215	0.12215	0.12215
	promedio	0.06671	0.06422	0.07066
	dsv. est.	0.03264	0.02424	0.02832
	peor	0.01732	0.03755	0.03810
Wine	mejor	0.52452	0.52452	0.52452
	promedio	0.42883	0.46231	0.47134
	dsv. est.	0.07907	0.04794	0.04892
	peor	0.27582	0.38914	0.36523
Zoo	mejor	0.83126	0.83126	0.83126
	promedio	0.64157	0.60371	0.61276
	dsv. est.	0.10967	0.14643	0.13300
	peor	0.53432	0.37828	0.44539
Wisconsin Breast Cancer	mejor	0.80669	0.80669	0.80669
	promedio	0.71310	0.69984	0.71996
	dsv. est.	0.05608	0.05647	0.04643
	peor	0.63608	0.60092	0.65515

Tabla 12: Estadísticas descriptivas para los promedios de los valores de aptitud obtenidos por los algoritmos implementados.

Debido a que *Apriori* no se encuentra dentro del grupo de modelos evolutivos, este no cuenta con la métrica de la función de aptitud, es por esta razón que para evaluar la calidad de las reglas generadas por este algoritmo, se toma en consideración el promedio de la confianza de las 10 mejores reglas obtenidas. Para la ejecución de *Apriori* se tomaron los umbrales de soporte y confianza mínimos obtenidos del modelo evolutivo que mostró mejores resultados. En [5] y [26] también se utilizó la confianza para estimar la calidad de las reglas de asociación obtenidas.

Base de datos	BPSO	BGWO	i-BGWO	BPSO [26]	BPSO [5]	A priori
Charles Book Club	0.51191	0.4945	0.51593	0.6951	-	0.35465
MovieLens	0.75832	0.75833	0.81621	-	-	0.71564
Foodmart 2000	0.38443	0.39040	0.41131	-	0.3996	0.38004
Wine	0.52942	0.54865	0.61715	-	-	0.51546
Zoo	0.69578	0.68002	0.69578	-	-	0.67546
Wisconsin Breast Cancer	0.80736	0.77882	0.79999	-	-	0.77546

Tabla 13: Valor de confianza promedio de las reglas obtenidas por los algoritmos.

BASE DE DATOS	ESTAD.	BPSO	BGWO	I-BGWO	A PRIORI
Charles Book Club	mejor	0.65506	0.65506	0.65506	0.63548
	promedio	0.51191	0.4945	0.51593	0.35465
	dsv. est.	0.11313	0.13263	0.10925	0.10056
	peor	0.38298	0.29866	0.38297	0.30546
MovieLens	mejor	1	1	1	1
	promedio	0.75832	0.75833	0.81621	0.71564
	dsv. est.	0.32024	0.32023	0.30987	0.30015
	peor	0.25000	0.25000	0.16216	0.17541
Foodmart 2000	mejor	0.42080	0.42349	0.46260	0.40564
	promedio	0.38443	0.39040	0.41131	0.38004
	dsv. est.	0.07005	0.05242	0.02213	0.02456
	peor	0.18869	0.24573	0.38581	0.19548
Wine	mejor	0.65263	0.80220	0.80220	0.54654
	promedio	0.52942	0.54865	0.61715	0.51546
	dsv. est.	0.09948	0.20087	0.08956	0.07897
	peor	0.34066	0.04819	0.50549	0.34568
Zoo	mejor	0.91667	0.80000	0.91667	0.80156
	promedio	0.69578	0.68002	0.69578	0.67546
	dsv. est.	0.12067	0.09712	0.12067	0.08456
	peor	0.52459	0.52459	0.52459	0.50879
Wisconsin Breast Cancer	mejor	0.92661	0.92661	0.92661	0.90584
	promedio	0.80736	0.77882	0.79999	0.77546
	dsv. est.	0.07559	0.10492	0.09746	0.07456
	peor	0.69841	0.63436	0.63436	0.63125

Tabla 14: Estadísticas descriptivas para los promedios de los valores de confianza obtenidos por los algoritmos implementados.

5.3. Caso de estudio enfocado al análisis de canasta de mercado con datos de un negocio local.

Para la base de datos de Mi Tiendita, descrita en la sección sección 4.3.1, se utilizó el *i-BGWO* y los otros dos algoritmos implementados en esta investigación *BPSO* y *BGWO* para minar reglas de asociación. Debido al gran tamaño de la base de datos se decidió generar las reglas de asociación por trimestre. Se excluyeron todos los ítems que no pasaran el umbral mínimo de 0.00259, lo que es igual a que el ítem apareciera por lo menos 30 veces en el conjunto de datos. En este trabajo se tomaron en consideración los ítems con mayor frecuencia de aparición en los conjuntos de datos, aunque existe una gran oportunidad para minar reglas de asociación seleccionando diferentes rangos de umbrales de soporte para encontrar reglas de asociación no comunes o raras, no obstante este método quedaría fuera de los alcances de la actual investigación.

Se realizó una comparación sobre la calidad de las reglas obtenidas con las implementaciones de *BGWO*, *i-BGWO* y *BPSO*. En la Tabla 15, se puede apreciar el promedio de la aptitud y confianza para las 10 mejores reglas generadas por los algoritmos. Los promedios de las métricas en la mayoría de los trimestres suelen ser mayores con el Optimizador del Lobo Gris Binario Mejorado *i-BGWO*, lo que significa que, si bien este algoritmo puede llegar a igualar y obtener mejores resultados a los que se obtienen con *BPSO*, el cual es uno de los algoritmos evolutivos basados en inteligencia de enjambres más utilizados en la literatura para la minaría de reglas de asociación (figura 6), el *i-BGWO* denota ser relevante, y proveer resultados competitivos con respecto a propuestas del estado del arte, al igual que ofrece una gran oportunidad de investigación, debido a su uso casi inexistente en el área. Los parámetros utilizados en los algoritmo fueron expuestos en la sección 5.1.

Promedio	Primer Trimestre		Segundo Trimestre	
	fitness	confianza	fitness	confianza
i-BGWO	0.003995	0.315933	0.007094	0.318204
BGWO	0.003622	0.311094	0.0063	0.365996
BPSO	0.002947	0.26694	0.006166	0.249809

Promedio	Tercer Trimestre		Cuarto Trimestre	
	fitness	confianza	fitness	confianza
i-BGWO	0.005381	0.332733	0.006642	0.342658
BGWO	0.003567	0.228249	0.005358	0.276965
BPSO	0.001174	0.159482	0.004013	0.351276

Tabla 15: Promedio de la función de aptitud y confianza de las reglas generadas con la base de datos Mi Tiendita.

APTITUD				
ESTAD.	BPSO	BGWO	I-BGWO	-
mejor	0.01227	0.01227	0.00784	-
promedio	0.00358	0.00471	0.00578	-
dsv. est.	0.00339	0.00318	0.00191	-
peor	0.00145	0.00145	0.00145	-

CONFIANZA				
ESTAD.	BPSO	BGWO	I-BGWO	A PRIORI
mejor	1	1	1	1
promedio	0.25688	0.29558	0.32738	0.25465
dsv. est.	0.14568	0.20412	0.20412	0.16546
peor	0.45610	0.38889	0.38889	0.35456

Tabla 16: Estadísticas descriptivas para los promedios de los valores de aptitud y confianza obtenidos por la base de datos de Mi Tiendita.

En los apéndices, se muestran las tablas correspondientes a las reglas de asociación generadas para los cuatro trimestres.

5.4. Prueba estadística Mann-Whitney-Wilcoxon.

Con el objetivo de comprobar si los resultados reportados para la base de datos del caso de estudio enfocado al análisis de canasta de mercado (Mi Tiendita) muestran o no una diferencia significativa entre ellos, se realizó la prueba *Mann-Whitney-Wilcoxon* [60]. Esta compara el rango medio de los datos proporcionados para las mejores iteraciones obtenidas por los algoritmos implementados en esta investigación.

Para realizar la prueba se utilizó el modulo de *Mannwhitneyu* de la librería *Scipy* en *Python 3.9*. En la tabla 17 se muestran los valores p obtenidos de *i-BGWO* contra los otros algoritmos implementados, tomando en cuenta una confianza del 95 %. Se definió como hipótesis nula H_0 que la diferencia entre los algoritmos no es significativa y como hipótesis alternativa H_1 , que la diferencia es significativa.

Valor de p para i-BGWO vs otros				
valor-p	Aptitud	i-BGWO	BPSO	0.39914
	Confianza		BGWO	0.35073
BPSO		0.37456		
BGWO		0.30464		
			Apriori	0.22164

Tabla 17: Valores de p obtenidos con la prueba Mann-Whitney-Wilcoxon.

Al ser todos los valores mayores a 0.05, puede afirmarse que se rechaza la hipótesis alternativa, por lo que la diferencia de los resultados obtenidos con los algoritmos no es significativa.

5.5. Evaluación del tiempo de ejecución de *i-BGWO*.

Para evaluar el tiempo de ejecución de los algoritmos implementados se utilizó la base de datos de Mi Tiendita, en esta solo se utilizaron las transacciones e ítems correspondientes al mes de enero de 2016, este periodo cuenta con un total de 11,578 transacciones y 1,497 ítems. Las métricas que competen a esta evaluación están condicionadas por el número de transacciones e ítems, de esta manera determinar la variable más influyente en el desempeño del algoritmo para generar reglas de asociación. Se optó por utilizar este conjunto de datos para medir el desempeño del tiempo de ejecución, debido a la gran cantidad de ítems y transacciones que posee.

5.5.1. Por número de ítems.

Con el objetivo de calificar el rendimiento de los algoritmos implementados en cuanto al tiempo de ejecución, se redujo el tamaño de la base de datos de 1,497 ítems a 149, lo que corresponde al 10 % del total de los elementos presentes en el periodo de enero de 2016. Adicionalmente, solo se tomaron en consideración los elementos que tuvieron una frecuencia de aparición de 30 veces en la base de datos reducida, lo que es igual a un umbral de soporte mínimo de 0.002591 para el respectivo periodo. En la figura 13 se puede apreciar el desempeño en tiempo de ejecución en segundos, que se obtuvo con las implementaciones de *BGWO*, *BPSO* y *i-BGWO* para la generación de las 10 mejores reglas. Cada punto en las líneas representa una regla y el valor que está sobre de estas son los segundos acumulados.

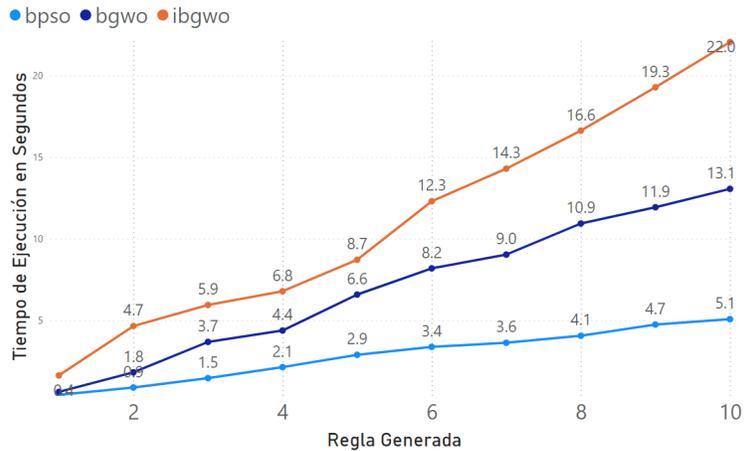


Figura 13: Tiempo de ejecución por cantidad de ítems.

5.5.2. Por número de transacciones.

Para calificar el rendimiento de los algoritmos implementados en cuanto al tiempo de ejecución por la cantidad de transacciones presentes en la base de datos, se redujo la cantidad original de 11,578 transacciones a 2,063, lo que corresponde al 17% del total de las transacciones presentes en el periodo de enero de 2016, para reducir la cantidad de transacciones en la base de datos, solo se tomaron en consideración los registros donde aparecen por lo menos 4 ítems. En la figura 14 se puede apreciar el desempeño que obtuvieron las implementaciones de *BGWO*, *BPSO* y *i-BGWO* con respecto al tiempo de ejecución en segundos (eje Y) y al número de transacciones para las 10 mejores reglas generadas (eje X).

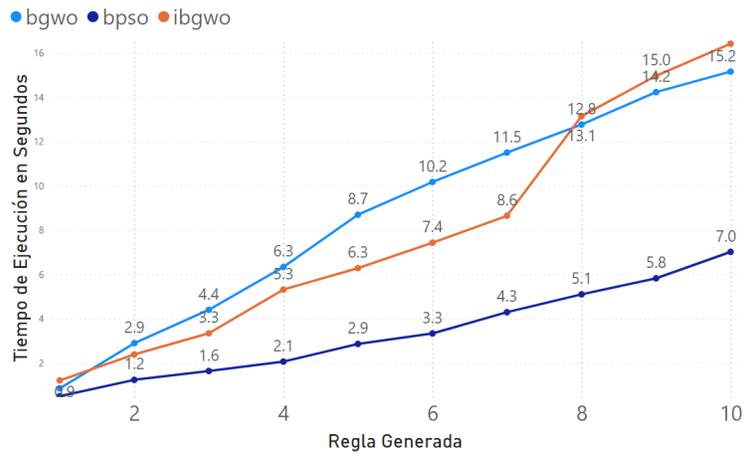


Figura 14: Tiempo de ejecución por cantidad de transacciones.

6. Conclusiones y trabajo futuro.

En este trabajo se propuso una modificación a la implementación el Optimizador del Lobo Gris Mejorado *i-GWO* propuesto en [14], para la tarea de minar reglas de asociación en bases de datos transaccionales. Las reglas de asociación son computacionalmente representadas por vectores binarios, haciéndolo un problema discreto. El optimizador del lobo gris fue originalmente diseñado para resolver problemas continuos, por lo que en este trabajo se adecuo una versión mejorada de este algoritmo presentada en [14], al dominio binario *i-BGWO*, para realizar la labor de minar reglas de asociación, algo que no había sido explorado con anterioridad. Para realizar la conversión al este dominio se siguió la metodología presentada en [33], donde el optimizador del lobo gris canónico fue transportado a su versión binaria.

El método propuesto en este trabajo fue comparado contra otros algoritmos pertenecientes al estado del arte. Para evaluar el desempeño se tomó en consideración la calidad y el tiempo de ejecución de las reglas generadas, para estimar la calidad se computó el promedio del valor de aptitud y de la confianza. En cuanto al tiempo de ejecución, se realizaron pruebas con una base de datos proveniente de una tienda de conveniencia, donde se evaluó al modelo acorde a la cantidad de transacciones y de ítems correspondientes al mes de enero.

Para estimar el desempeño del algoritmo propuesto en cuanto a la calidad de las reglas generadas, se tomaron los promedios de la aptitud y confianza de las 10 mejores reglas generadas, obtenidas con los modelos implementados en este trabajo y con los resultados obtenidos en otras publicaciones del área. Los resultados muestran que en la mayor parte del tiempo, las reglas generadas con el Optimizador del Lobo Gris Mejorado Binario *i-BGWO* tienen mayor calidad al tener un valor más alto, debido a la naturaleza intrínseca de maximización de la función de aptitud. Con los resultados expuestos en la sección anterior, se puede corroborar que el *i-BGWO* obtiene los promedios más altos para las M mejores reglas (parámetro/variable de entrada) la mayor parte del tiempo, por lo que la utilización de este modelo para la tarea de minar reglas de asociación denota ser una aportación significativa para el área.

En cuanto al tiempo de ejecución, se evaluó al *i-BGWO* respecto al número de ítems y transacciones pertenecientes al caso de estudio de la tienda de conveniencia, con los datos correspondientes al periodo de enero

de 2016. En las figuras 13 y 14 se muestra en el eje Y el tiempo de ejecución en segundos y en el eje X el número de la regla generada. Estos resultados muestran que el *i-BGWO* llega a ser más costoso en cuanto al tiempo de ejecución comparado con el *BGWO* canónico y el *PSO* binario, debido al paso adicional que el *i-GWO* implementa al momento de generar la nueva posición candidata a través de la creación del vecindario, con la técnica *DLH*.

Con la utilización de la prueba no paramétrica *Mann-Whitney-Wilcoxon* [60] se pudo constatar que los resultados a pesar de favorecer la mayor parte del tiempo al *i-BGWO*, no muestran una diferencia significativa entre los valores obtenidos por los algoritmos implementados en esta investigación para la base de datos enfocada al caso de estudio del análisis de canasta de mercado.

Con los resultados obtenidos en las diferentes pruebas de rendimiento, se pueden resaltar ventajas como desventajas de utilizar el algoritmo propuesto para minar reglas de asociación, la aportación más importante que ofrece *i-BGWO*, es la generación de reglas de asociación de mayor calidad, haciendo de este una opción competitiva en el área, aunque esta propuesta llega a ser costosa en cuanto al tiempo de ejecución, debido al paso adicional que implementa (*DLH*).

Estas afirmaciones dan cumplimiento a la hipótesis propuesta en esta investigación, la cual menciona que el *i-BGWO* es una alternativa eficiente para minar reglas de asociación de mayor calidad para un problema de estudio específico, al igual que ser una opción competitiva contra otros modelos del estado del arte en esta temática.

En [5] se propuso la implementación del *itemset range*, el cual codifica el vector con los ítems de cada transacción, seleccionando al antecedente y al consecuente dependiendo del valor obtenido para un límite superior y uno inferior para cortar el vector. Esta selección está basada en un cálculo que toma en consideración las métricas y propiedades de cada transacción, lo que difiere de la metodología propuesta en este trabajo para codificar las transacciones, donde el antecedente y el consecuente son elegidos de manera aleatoria. Esta estrategia podría traer mejoras en la implementación del *i-BGWO* para el minado de reglas de asociación.

Las métricas más comunes para evaluar a las reglas de asociación son el soporte y la confianza, aunque en la minería de reglas de asociación también

se ha propuesto la métrica de levantamiento (*Lift*, por su nombre en inglés, para evaluar la calidad de las reglas generadas. Esta métrica explica el incremento en la proporción de aparición de un ítem cuando otro está presente en la transacción. Esta métrica no se ha utilizado en los trabajos relacionados con la computación evolutiva para la minería de reglas de asociación, por lo que podría ser de utilidad para tener otra perspectiva al momento de evaluar las reglas de asociación generadas por modelos evolutivos.

En este trabajo se optó por utilizar un rango de umbral de soporte que abarcara solo a los ítems más frecuentes, aunque también existe una gran oportunidad para minar reglas de asociación seleccionando diferentes rangos de umbrales de soporte, para encontrar reglas de asociación no comunes o raras, que no se encuentren dentro de los *conjuntos de datos* frecuentes y que puedan ser de interés para un caso de estudio específico.

7. Apéndices.

BPSO - Primer Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[75001322, 75022907]	[75027667]	0.013383	0.005665	0.794118
2	[75001322, 75027667]	[75022907]	0.006894	0.005665	0.409091
3	[7501032346324, 7502217040556]	[7501013174021]	0.002919	0.002308	0.423077
4	[7501018310103, 7501158414310]	[7501005108102]	0.00253	0.004406	0.192661
5	[7501005180306, 7502217040556]	[735257000076]	0.001099	0.001469	0.25
6	[7501073831469, VASOS]	[11313-0027]	0.000827	0.001049	0.263133
7	[75001322, 7502217040556]	[7501055310227]	0.000686	0.002937	0.078212
8	[ECOCA2.5LT, VASOS]	[7501055310227]	0.000592	0.002728	0.072626
9	[823703800193, HB2]	[034587030013]	0.000481	0.001259	0.12766
10	[7501000138944, 7501055308675, 7503011952045]	[7501032346362]	0.000062	0.00021	0.058824
BGWO - Primer Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[75001322, 75022907]	[75027667]	0.013383	0.005665	0.794118
2	[75001322, 75027667]	[75022907]	0.006894	0.005665	0.409091
3	[7501147515769, HB2]	[HIELO]	0.003438	0.003147	0.365854
4	[AZUCAR, HB2]	[823703800193]	0.003119	0.004826	0.216981
5	[7501032346324, 7502217040556]	[7501013174021]	0.002919	0.002308	0.423077
6	[7501018310103, 7501158414310]	[7501005108102]	0.00253	0.004406	0.192661
7	[HB2, QUESO1]	[HUASTECO]	0.002207	0.002728	0.270833
8	[7501073830509, VASOS]	[11313-0003]	0.000714	0.001049	0.227273
9	[823703800193, HB2]	[7501379119056]	0.000598	0.002518	0.07947
10	[7501055303540, VASOS]	[7501011132283]	0.000413	0.001049	0.131579
i-BGWO - Primer Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[75001322, 75022907]	[75027667]	0.013383	0.005665	0.794118
2	[75001322, 75027667]	[75022907]	0.006894	0.005665	0.409091
3	[7501055310227, VASOS]	[ECOCA2.5LT]	0.005045	0.002728	0.619048
4	[823703800193, HB2]	[7501431201088]	0.003232	0.002937	0.368421
5	[AZUCAR, HB2]	[823703800193]	0.003119	0.004826	0.216981
6	[7501379120045, 823703800193, AZUCAR]	[7501379144645]	0.002681	0.002937	0.229508
7	[7501018310103, 7501158414310]	[7501005108102]	0.00253	0.004406	0.192661
8	[7503015181021, HUEVO]	[7501055901128]	0.001855	0.003147	0.197368
9	[HB2, HIELO]	[7501147515769]	0.001029	0.003147	0.109489
10	[7501147515769, 7502223770805]	[7503011952045]	0.000185	0.002728	0.022648

Tabla 18: 10 mejores reglas de asociación generadas, correspondientes al primer trimestre de 2016.

BPSO - Segundo Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[7501055901104, 7501055901128]	[7501055901111]	0.018408	0.015205	0.412698
2	[75001322, 75027667]	[75022907]	0.013865	0.008187	0.571419
3	[7501055901111, 7501055901128]	[7501055901104]	0.013104	0.015205	0.293792
4	[7501007704005, 758104100422]	[097339000054]	0.004606	0.003801	0.40625
5	[7501055901128, 7501055901135]	[7501055901111]	0.004304	0.00731	0.198413
6	[75022907, 75027667]	[75001322]	0.003817	0.008187	0.157303
7	[7501055901111, 7501055901159]	[7501055901128]	0.001783	0.00614	0.097674
8	[7501059284555, 7502217040556]	[7501001600105]	0.00075	0.001754	0.142857
9	[7502217040556, HB2]	[7501055905287]	0.000729	0.001462	0.166667
10	[7501055901111, 7501055901128, 7501055901135]	[7501055901159]	0.000297	0.001462	0.05102
BGWO - Segundo Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[75001322, 75022907]	[75027667]	0.021916	0.008187	0.903226
2	[7501055901104, 7501055901128]	[7501055901111]	0.018408	0.015205	0.412698
3	[7501059284555, 7502217040556]	[735257005019]	0.006613	0.003509	0.631579
4	[7501007704005, 758104100422]	[097339000054]	0.004606	0.003801	0.40625
5	[7501055901128, 7501055901135]	[7501055901111]	0.004304	0.00731	0.198413
6	[7501055310227, VASOS]	[ECOCA2.5LT]	0.003996	0.002339	0.571429
7	[7501055901128, HUEVO]	[7503015181021]	0.001653	0.003509	0.157895
8	[7503011952045, HUEVO]	[7501147929124]	0.000589	0.002339	0.084211
9	[7501059284555, 7501379119056, 7502217040556]	[735257005019]	0.000553	0.000877	0.157895
10	[7501158414310, 7501379119056]	[7501018310738]	0.000358	0.000877	0.136364
i-BGWO - Segundo Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[7501055901104, 7501055901128]	[7501055901111]	0.018408	0.015205	0.412698
2	[75001322, 75027667]	[75022907]	0.013865	0.008187	0.571429
3	[7501055901111, 7501055901128]	[7501055901104]	0.013104	0.015205	0.293785
4	[7501059284555, 7502217040556]	[735257005019]	0.006613	0.003509	0.631579
5	[7501055901104, 7501055901128]	[7501055901159]	0.005102	0.007018	0.244898
6	[7501007704005, 758104100422]	[097339000054]	0.004606	0.003801	0.40625
7	[75022907, 75027667]	[75001322]	0.003817	0.008187	0.157303
8	[7501055901128, 7501055901159]	[7501055901111]	0.003042	0.00614	0.166667
9	[7501055901128, HUEVO]	[7503015181021]	0.001653	0.003509	0.157895
10	[7501158414310, 7501379119056]	[7501018310523]	0.000732	0.001754	0.139535

Tabla 19: 10 mejores reglas de asociación generadas, correspondientes al segundo trimestre de 2016.

BPSO - Tercer Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[7501018310103, 7501158414310]	[7501005108102]	0.003044	0.005112	0.2
2	[7501055310227, ECOCA2.5LT]	[7501040009020]	0.001672	0.002236	0.249953
3	[7501000167456, 7501055328345]	[75001339]	0.001258	0.001597	0.263158
4	[7501055304745, HIELO]	[7501055304813]	0.001229	0.001917	0.214286
5	[ECOCA2.5LT, VASOS]	[7501055310227]	0.001142	0.004153	0.092199
6	[7501055902378, HB2]	[7501055909131]	0.001043	0.001917	0.181818
7	[7501055310227, ECOCA2.5LT]	[VASOS]	0.000728	0.004153	0.058824
8	[7501059238329, 7502217040556]	[7506192505451]	0.000589	0.001278	0.153846
9	[7501073800908, HIELO]	[7501055304745]	0.000568	0.003195	0.059526
10	[7501040000201, VASOS]	[7501055305247]	0.000464	0.001278	0.121212
BGWO - Tercer Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[75001322, 75022907]	[75027667]	0.021142	0.007987	0.892857
2	[7501055900800, 7502217040556]	[7501045401423]	0.003722	0.003514	0.354839
3	[75022907, 75027667]	[75001322]	0.003462	0.007987	0.146199
4	[7501059284555, 7501158414310]	[7501001600105]	0.002884	0.003514	0.275
5	[7501005108102, 7501158414310]	[7501018310103]	0.002081	0.005112	0.136752
6	[HORNO, VASOS]	[7501055303540]	0.000882	0.001917	0.153846
7	[7502217040556, HB2]	[034587030013]	0.00052	0.001597	0.108696
8	[7503011952045, HORNO]	[7501055310227]	0.000433	0.002556	0.056738
9	[7501055305247, VASOS]	[7501040000201]	0.000373	0.001278	0.097561
10	[7501017375622, 7506205802133]	[7503015181021]	0.000172	0.000958	0.06
i-BGWO - Tercer Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[75001322, 75022907]	[75027667]	0.021142	0.007987	0.892857
2	[75001322, 75027667]	[75022907]	0.014799	0.007987	0.625
3	[7501055310227, VASOS]	[ECOCA2.5LT]	0.006439	0.004153	0.52
4	[75022907, 75027667]	[75001322]	0.003462	0.007987	0.146199
5	[7501040009020, 7501055310227]	[ECOCA2.5LT]	0.001872	0.002236	0.28
6	[7502217040556, QUESO2]	[7501055900800]	0.001445	0.000319	0.076923
7	[7501147515769, 7501158414310]	[7501060500019]	0.001328	0.002556	0.173913
8	[7502217040556, HB2]	[7501026004612]	0.001229	0.001917	0.214286
9	[7501055308675, 7501055328345]	[7501055305339]	0.001132	0.002556	0.148148
10	[7501032346324, 7502217040556]	[7501013174021]	0.000957	0.001278	0.25

Tabla 20: 10 mejores reglas de asociación generadas, correspondientes al tercer trimestre de 2016.

BPSO - Cuarto Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[75001322, 75022907]	[75027667]	0.02191	0.008185	0.903226
2	[75001322, 75027667]	[75022907]	0.017416	0.008185	0.717949
3	[7501055310227, VASOS]	[ECOCA2.5LT]	0.006716	0.002923	0.769199
4	[7501013174021, 7502217040556]	[7501032346324]	0.003635	0.004092	0.297872
5	[7501018310103, 7501158414310]	[7501005180306]	0.00116	0.003215	0.120879
6	[75001322, 7501147929124, 7506195196984]	[75021597]	0.000982	0.001169	0.210526
7	[7503015181021, HUEVO]	[7501055901128]	0.0009	0.001754	0.171429
8	[75007614, 758104003921]	[7501073839885]	0.000493	0.000877	0.1875
9	[HB2, QUESO1]	[7501055901425]	0.000267	0.001754	0.050847
10	[089826001415, 7501055310227, HUEVO]	[7501061696988]	0.000097	0.000292	0.083333
BGWO - Cuarto Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[75001322, 75027667]	[75022907]	0.017416	0.008185	0.717949
2	[7501055303540, VASOS]	[7501011112254]	0.010426	0.007308	0.480769
3	[75001322, 7501147929124]	[75021597]	0.003724	0.002631	0.473684
4	[75022907, 75027667]	[75001322]	0.00325	0.008185	0.133971
5	[7501018310103, 7501158414310]	[7501032600686]	0.001947	0.002046	0.318182
6	[7501055304745, HIELO]	[7501073800908]	0.001912	0.002631	0.243243
7	[7501057710360, 7502217040556]	[7501000111206]	0.000572	0.001754	0.109091
8	[7503015181021, HB2]	[7501063200015]	0.000415	0.000877	0.157895
9	[7501059284555, HB2]	[058897091687]	0.000311	0.001169	0.088889
10	[7501431261105, 7502217040556]	[7501000658923]	0.000161	0.001169	0.045977
i-BGWO - Cuarto Trimestre					
id	antecedente	consecuente	fitness	soporte	confianza
1	[75001322, 75022907]	[75027667]	0.02191	0.008185	0.903226
2	[75001322, 75027667]	[75022907]	0.017416	0.008185	0.717949
3	[7501055303540, VASOS]	[7501011112254]	0.010426	0.007308	0.480769
4	[7501055310227, VASOS]	[ECOCA2.5LT]	0.006716	0.002923	0.080645
5	[7501013174021, 7502217040556]	[7501032346324]	0.003635	0.004092	0.297872
6	[7501055304745, 7501055304820]	[7501055304790]	0.00306	0.002046	0.5
7	[7501055304745, HIELO]	[7501073800908]	0.001912	0.002631	0.243243
8	[ECOCA2.5LT, VASOS]	[7501055310227]	0.000704	0.002923	0.080645
9	[7501011161207, VASOS]	[7501055302925]	0.000359	0.001169	0.102564
10	[7503011952045, QUESO2]	[QUESO1]	0.000284	0.002046	0.019663

Tabla 21: 10 mejores reglas de asociación generadas, correspondientes al cuarto trimestre de 2016.

Referencias

- [1] Shearer, C.: *The CRISP-DM Model: The New Blueprint for Data Mining*. Journal of Data Warehousing, 5, 13-22., 2000.
- [2] Adamo, J. M.: *Data mining for association rules and sequential patterns: sequential and parallel algorithms*. Springer Science Business Media, 2012.
- [3] Ansel Yoan Rodríguez González, José Francisco Martínez Trinidad, Jesús Ariel Carrasco Ochoa José Ruiz Shulcloper: *Minería de Reglas de Asociación sobre Datos Mezclados*. Coordinación de Ciencias Computacionales - INAOE, 2009.
- [4] Akbar T, Amir H, Asadollah S.: *A survey of evolutionary computation for association rule mining*. ELSEVIER, 2020.
- [5] R.J. Kuo, , C.M. Chao b Y.T. Chiuc: *Application of particle swarm optimization to association rule mining*. Applied Soft Computing, ELSEVIER, 2011.
- [6] Aggarwal, Charu C.: *Data Mining - The Textbook*. Springer International Publishing AG Switzerland, 2015.
- [7] Larose, D. T.: *Discovering knowledge in data: an introduction to data mining*. Discovering knowledge in data: an introduction to data mining., 2014.
- [8] Rakesh Agrawal, Tomasz Imielinski, Arun Swami: *Mining Association Rules between Sets of Items in Large Databases*. IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, 1993.
- [9] Aggarwal, Charu C.: *Recommender Systems - The Textbook*. Springer International Publishing AG Switzerland, 2016.
- [10] Julio C, Guillermo M, Carlos M María A.: *Reducción de tamaño en Modelos de Reglas de Asociación: Una revisión sistemática de la literatura*. Revista Cubana de Ciencias Informáticas Vol. 15, No. 3, 2021.
- [11] J. Han, H. Pei y Y. Yin.: *Mining Frequent Patterns without Candidate Generation*. Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA, 2000.

- [12] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay C.A.C. Coello: *A survey of multiobjective evolutionary algorithms for data mining: Part I*. IEEE Trans. Evol. Comput. 18 (1) 4–19, 2014.
- [13] S. Mirjalili, S. M. Mirjalili y A. Lewis: *Grey wolf optimizer*. Advances in Engineering Software, vol. 69, pp. 46–61, 2014.
- [14] Mohammad H. Nadimi-Shahraki, Shokooh Taghian, Seyedali Mirjalili: *An improved grey wolf optimizer for solving engineering problems*. Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Australia - Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran - Yonsei Frontier Lab, Yonsei University, Seoul, Korea. 2020.
- [15] Azevedo, A. I. R. L. y M. F. Santo: *Kdd, semma and crisp-dm: a parallel overview*. IADS-DM, 2008.
- [16] Gupta, S. y R. Mamtora: *A survey on association rule mining in market basket analysis*. International Journal of Information and Computation Technology, ISSN, pp. 0974–2239., 2014.
- [17] H. Orallo, J. RAMIREZ, C. R. QUINTANA M. J. H. Orallo M. J. R. Quintana: *Introducción a la Minería de Datos*. Pearson Prentice Hall, 2004.
- [18] R. Agrawal, R. Srikant, et al.: *Fast algorithms for mining association rules*. Proc. 20th int. conf. very large data bases, VLDB, vol. 1215, pp. 487–499, 1994.
- [19] J. S. Park, M. S. Chen y P. S. Yu: *Using a hash-based method with transaction trimming for mining association rules*. IEEE transactions on knowledge and data engineering, vol. 9, no. 5, pp. 813–825, 1997.
- [20] Holt, J. D. y S. M. Chung.: *Mining association rules using inverted hashing and pruning*. Information Processing Letters, vol. 83, no. 4, pp. 211–220., 2002.
- [21] Pietracaprina, A.: *Mining frequent itemsets using patricia tries*. 2003.
- [22] Sucahyo, Y. G. y R. P. Gopalan.: *Ct-itl: Efficient frequent item set mining using a compressed prefix tree with pattern growth*. Proceedings of the 14th Australasian database conference-Volume 17, pp. 95–104, Australian Computer Society, Inc., 2003.

- [23] Sucahyo, Y. G. y R. P. Gopalan.: *Ct-pro: A bottom-up non recursive frequent itemset mining algorithm using compressed fp-tree data structure*. FIMI, vol. 4, pp. 212–223., 2004.
- [24] A. Vikhar., Pradnya: *Evolutionary Algorithms: A Critical Review and its Future Prospects*. International Conference on Global Trends in Signal Processing, Information Computing and Communication, 2016.
- [25] S. Muppidi, M. R. Murty y S. Ch.: *Efficient frequent pattern mining using particle swarm optimization*. International Journal of Applied Engineering Research, vol. 12, no. 16, pp. 5520–5524, 2017.
- [26] Sarath, K. y V. Ravi.: *Association rule mining using binary particle swarm optimization*. Engineering Applications of Artificial Intelligence, vol. 26, no. 8, pp. 1832–1840, 2013.
- [27] Y. Gao, J. Q. Hu y X. L. Tang: *The application of hybrid ant colony algorithm in association rule mining*. Computational Intelligence and Design (ISCID), 2016 9th International Symposium on, vol. 2, pp. 329–333, IEEE., 2016.
- [28] al., C. Kant et: *Association rule mining using ant colony optimization*. 2015.
- [29] P. Sharma, S. Tiwari y M. Gupta: *A novel approach for reducing the candidate item sets and large item sets by fuzzy mining association rule*. Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on, pp. 370–373, IEEE, 2015.
- [30] R. Miri, A. Agrawal, A. Miri y S. Tandan.: *Association rule mining using ant colony optimization*. Fuzzy Systems, vol. 8, no. 5, pp. 126–128, 2016.
- [31] Hanirex, D. K. y K. Kaliyamurthie.: *Mining frequent itemsets using genetic algorithm*. Middle-East Journal of Scientific Research, vol. 19, no. 6, pp. 807–810, 2014.
- [32] D. Martin, A. Rosete, J. Alcalá Fdez y F. Herrera: *A new multiobjective evolutionary algorithm for mining a reduced set of interesting positive and negative quantitative association rules*. IIIE., 2014.
- [33] E. Emary, Hossam M. Zawbaa, Aboul Ella Hassanien: *Binary Gray Wolf Optimization Approaches for Feature Selection*. ELSEVIER, 2016.

- [34] Hossein N., Majid R., Malihe M.: *Binary Particle Swarm Optimization: Challenges and New Solutions*. Department of Electrical Engineering, Shahid Bahonar University of Kerman , P.O. Box 76169-133, Kerman, Iran., 2008.
- [35] Aggarwal, C. y J. Han.: *Frequent pattern mining*. Springer International Publishing AG Switzerland, New York, 2014.
- [36] Bastide, Yves: *Mining minimal non-redundant association rules using frequent closed itemsets*. Springer, Berlin, Heidelberg., 2000.
- [37] Tirnauca, C., Balcazar J. L. Gomez Perez D.: *Closed-Set-Based Discovery of Representative Association Rules*. International Journal of Foundations of Computer Science, 31(1): p. 143 – 156, 2020.
- [38] B. Alatas, E. Akin: *Multi-objective rule mining using a chaotic particle swarm optimization algorithm*. Knowl.-Based Syst. 22 (6) 455–460, 2009.
- [39] W. Wang, S. Bridges: *Genetic Algorithm Optimization of Membership Functions for Mining Fuzzy Association Rules*. Department of Computer Science Mississippi State University, 2000.
- [40] B. Alatas, E. Akin, A. Karci: *MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules*. Appl. Soft Comput. 8 (1) 646–656, 2008.
- [41] Maliha, A. Sarowar, Mahit K.: *Market Basket Analysis Using Apriori and FP Growth Algorithm*. 22nd International Conference on Computer and Information Technology (ICIT), 2019.
- [42] Wiwit P, Hanipar M, Abdul R Muhammad G. Ristu H Zuhut R Muhammad M.: *Market Basket Analysis with Apriori Algorithm and Frequent Pattern Growth (Fp-Growth) on Outdoor Product Sales Data*. International Journal of Educational Research Social Sciences, 2021.
- [43] Della Maulina Herianda, Ahmad I, Achmad D Folkes E Nuning K Akbar I Gloria M Ida B Robbi R.: *Market Basket Analysis Using Apriori and FP-Growth for Analysis Consumer Expenditure Patterns at Berkah Mart in Pekanbaru Riau*. IOP Conf. Series: Journal of Physics, 2018.
- [44] Alexandros T, Georgios D.: *A Comprehensive Survey on the Applications of Swarm Intelligence and Bio-Inspired Evolutionary Strategies*.

- Springer, Machine Learning Paradigms - Advances in Deep Learning-based Technological Applications (Chapter 15) 337–378, 2020.
- [45] Jr., Iztok Fister y Iztok Fister: *A brief overview of swarm intelligence-based algorithms for numerical association rule mining*. 2020.
- [46] P., Madhusmita y Bikramaditya D.: *Grey Wolf Optimizer and Its Applications: A Survey*. Proceedings of the Third International Conference on Microelectronics, Computing and Communication Systems. 179–194, 2019.
- [47] Parsian A, Ramezani M, Ghadimi N: *A hybrid neural network-Gray Wolf optimization algorithm for melanoma detection*. Biomed Res 28(8), 2017.
- [48] Zhang S, Zhou Y, Li Z Pan W.: *Grey Wolf optimizer for unmanned combat aerial vehicle path planning*. Adv Eng Softw 99:121–136, 2016.
- [49] S. Sharmila, S.Vijayarani.: *Association Rule hiding Using Grey Wolf Optimization Algorithm*. International Journal of Engineering and Advanced Technology (IJEAT), 2019.
- [50] N. Pazhaniraja, S. Sountharajan, B. Sathis Kumar: *High utility item-set mining: a Boolean operators-based modified grey wolf optimization algorithm*. Soft Computing, Springer, 2020.
- [51] Gungor Yildirim1, Bilal Alatas: *New adaptive intelligent grey wolf optimizer based multi-objective quantitative classification rules mining approaches*. Journal of Ambient Intelligence and Humanized Computing, Springer, 2020.
- [52] O. M. Badawy, A. E. A. Sallam y M. I. Habib: *Quantitative association rule mining using a hybrid pso/aco algorithm*. 2008.
- [53] al., Gretel Bernal et: *A PSO-based algorithm for mining association rules using a guided exploration strategy*. Pattern Recognition Letters, ELSEVIER, 2020.
- [54] *Charles Book Club*. <https://github.com/prnvg/Charles-Book-Club>.
- [55] *MovieLens*. <https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset>.

- [56] *Foodmart 2000*. https://documentation.help/dtsprog/dtsp_samp_7smo.htm.
- [57] Garg, Anisha: *UCI Machine Learning Repository*. <https://archive.ics.uci.edu/ml/datasets.php>.
- [58] Xilot, Jaime Elí Hernández: *Modelo basado en datos históricos para la predicción de venta nula de productos en tiendas de conveniencia*. Laboratorio Nacional de Informatica Aplicada, Febrero 2018.
- [59] Hisao Ishibuchi, Tomoharu Nakashima y Tadahiko Murata: *Comparison of the Michigan and Pittsburgh Approaches to the Design of Fuzzy Classification Systems*. College of Engineering, Osaka Prefecture University, Sakai, Osaka, Japan 593, 1997.
- [60] Mann, H. B. y D. R. Whitney: *On a test of whether one of two random variables is stochastically larger than the other*. Ann. Math. Stat. 18: 50-60, 1947.
- [61] Garg, Anisha: *Complete guide to Association Rules*, September 2018. <https://towardsdatascience.com/association-rules-2-aa9a77241654>.
- [62] Mezura-Montes, Efrén y Carlos A. Coello Coello: *Useful Infeasible Solutions in Engineering Optimization with Evolutionary Algorithms*. Computer Science Section, Electrical Engineering Department, Av. IPN No. 2508 Col, San Pedro Zacatenco Mexico D.F. 07300, Mexico.
- [63] R. Agrawal, R. Srikant, et al.: *Fast algorithms for mining association rules*. Proc. 20th int. conf. very large data bases, VLDB, vol. 1215, pp. 487-499, 1994.
- [64] McNicholas, P.D., Murphy T.B. O'Regan M.: *Standardising the lift of an association rule*. ELSEVIER, 2008.