Universidad Central “Marta Abreu” de Las Villas

Facultad de Matemática, Física y Computación

Centro de Estudios de Informática

Algoritmo de selección de rasgos en fuentes de datos distribuidas.

Autor: Enrique Alfonso Casanovas Pedre

Tutores: Lic. Wilver Díaz Bello

Dr. Amílkar Puris Cáceres

Dedicatoria

A mi familia y amigos.

Agradecimientos

Muchas gracias a todas las personas
que me ayudaron a llegar hasta aquí.

A mis padres, a mi hermanita y a mi abuela
por la educación y el amor que me dieron.
A toda mi familia
por apoyarme siempre y darme ánimos para estudiar.

A los maestros y profesores ejemplares que siempre tuve,
a mi primera maestra de la escuela y excelente educadora María Orquídea Guevara, y
especialmente a mi profesor de física Filgueiras, por mostrarme qué es ser un maestro.

Muchas gracias a todos mis compañeros de la escuela y la universidad,
A los compañeros del Laboratorio de Inteligencia Artificial, profesores, tutores y estudiantes.
Un agradecimiento especial a mi tutor Yudel Gómez Díaz,
Por haber sido también mi amigo y por haber sido mi guía en el mundo de la investigación científica.

Por último quiero agradecer a Vicious Byte, Blizzard, Bioware, Ubisoft, Nintendo, Square Enix, Electronic Arts, Bethesda, Black Isle, Microsoft, Capcom, Marvel, id y Activision, por haber desarrollado ideas tan ingeniosas, ayudarme a entrenar mi imaginación, a combatir el aburrimiento y por cultivar mi amor por la programación y la inteligencia artificial.

A todos mis amigos, que no nombro porque sé que se me quedará alguno.

A todo el que lea este trabajo.

SÍNTESIS

El *problema de selección de rasgos* consiste en la elección de los atributos que mejor clasifican a los objetos de un dominio determinado; una de las fases más importantes del proceso conocido como *Descubrimiento de Conocimiento en Bases de Datos* (KDD). En muchos dominios de aplicación, los datos se encuentran en múltiples fuentes aisladas unas de otras, y por diversas razones, no es posible unirlos en una única fuente de datos. Para el estudio de este problema fue creada la *Minería de Datos Distribuida* (DDM), perteneciente también al KDD, y a la selección de rasgos en dicho contexto se le llama *selección de rasgos en fuentes de datos distribuidas*. Esta es un área de la ciencia que no ha sido ampliamente estudiada y a la que no se le han encontrado soluciones definitivas.

Esta tesis se presenta un enfoque para resolver el problema de selección de rasgos utilizando la Teoría de Conjuntos Aproximados, la Teoría de la Información y la Optimización basada en Colonias de Hormigas y se extiende dicho enfoque para realizar la selección en múltiples fuentes de datos mediante un mecanismo de intercambio de metadatos. Se realiza además una validación de los modelos propuestos utilizando seis fuentes de datos del Repositorio UCI y se comparan los resultados obtenidos con los de otros modelos presentes en la bibliografía.

ABSTRACT

The feature selection problem consists in selecting the attributes that better classify the objects of a given domain; one of the most important stages of the process known as Knowledge Discovery in Databases (KDD). In many application domains, data is found in multiple isolated sources one from each other, and because of different reasons it is not possible to join them all together in a single data source. Towards the study of this problem the *Distributed Data Mining* (DDM) was created, which belongs to KDD, and feature selection on this context is called *feature selection on distributed data sources*. This is an area of science with no definite solutions and which has not been widely studied.

In this thesis it is presented an approach to solve the feature selection problem using Rough Set Theory, Information Theory and Ant Colony Optimization and such approach is extended to achieve the selection in multiple data sources by means of a metadata interchange mechanism. It is also done a validation of the proposed models using six data sources from the UCI Repository and the obtained results are compared with those from other models in the bibliography.

Índice de contenidos

SÍNTESIS III
ABSTRACT IV
INTRODUCCIÓN 1
Problema de investigación 2
Objetivo general 2
Objetivos específicos 2
Preguntas de la investigación 2
Hipótesis 2
Justificación 3
1 La selección de rasgos y sus componentes 4
1.1 Descubrimiento de conocimiento, Minería de Datos y Minería de Datos Distribuida. 4
Contexto distribuido 5
1.2 Selección de rasgos 6
Reducto 8
1.3 Teoría de Conjuntos Aproximados 9
Definiciones principales de la Teoría de Conjuntos Aproximados 10
1.4 Teoría de la Información 14
Entropía 14
1.5 Optimización basada en colonias de hormigas 17
Diversas variantes de la familia de algoritmos basados en colonias de hormigas 21
1.6 Consideraciones parciales 24
2 Solución al problema de selección de rasgos 26
2.1 Propuestas para la solución de la selección de rasgos 26
2.2 Algoritmo ACO-RST-IT-FSP 27
Grafo de feromonas 28
Construcción de las hormigas y asignación inicial 28
Procedimiento de transición 29
Criterio de parada de las hormigas 29
Búsqueda local 30
Actualización fuera de línea de la feromona 30

Criterio de parada de la colonia 30
2.3 Análisis de la complejidad computacional de ACO-RST-IT-FSP 32
Estrategia de reducción del tiempo de ejecución 33
2.4 Solución al problema distribuido de selección de rasgos 35
2.5 Algoritmo D.ACO-RST-IT-FSP 36
2.6 Consideraciones parciales 38
3 Implementación y evaluación de los algoritmos 40
3.1 Validación del algoritmo ACO-RST-IT-FSP 42
3.2 Validación del algoritmo D.ACO-RST-IT-FSP 45
3.3 Conclusiones del análisis de los resultados 51
3.4 Consideraciones parciales 51
CONCLUSIONES 53
RECOMENDACIONES 54
REFERENCIAS 55
Anexos 60
Anexo 1 Características de los conjuntos de datos 60
Anexo 2. Resultados experimentales de la estrategia de ahorro 61
Anexo 3 Comparaciones entre varios algoritmos en contexto local 64

INTRODUCCIÓN

Durante los últimos años la humanidad se ha visto abrumada por los datos. La llegada de las computadoras personales y las nuevas tecnologías de la información y las comunicaciones han generado un incremento considerable en la cantidad de datos que se almacenan y se transmiten, siendo sólo una pequeña parte de esta verdaderamente comprendida por el hombre. Tal crecimiento en los volúmenes de datos, ha hecho surgir la necesidad de crear nuevas técnicas para extraer información útil de ellos.

Una vía para aplacar los inconvenientes causados por tanta cantidad de datos y facilitar su estudio consiste en la reducción de la dimensionalidad de los datos, de manera que solo quede la parte de los datos que es relevante, es decir, que aporta algún tipo de conocimiento. La selección de rasgos se enmarca en esta compleja temática, y es un problema aún abierto de la ciencia de la computación al que no se le ha dado solución completamente.

Debido a la gran complejidad de muchos problemas de la ciencia, entre ellos, la selección de rasgos, los métodos tradicionales no son capaces de solucionarlos eficientemente, por esta razón se ha recurrido a la creación de modelos inspirados en soluciones que la naturaleza le da a problemas similares. Uno de estos modelos bioinspirados es la *Optimización basada en Colonias de Hormigas*, la cual, combinada con teorías de análisis de datos, como la Teoría de Conjuntos Aproximados y la Teoría de la Información permite construir algoritmos capaces de solucionar, con resultados satisfactorios, el problema de la selección de rasgos.

En muchas ramas del desarrollo los datos se encuentran dispersos en diferentes nodos, ya sea geográfica, física o virtualmente y en muchos casos es imposible o no es factible unirlos en un único repositorio antes de proceder a su análisis, trayendo esto consigo una menor comprensión del fenómeno que representan. Por este motivo es necesario extender las ideas de los modelos bioinspirados y crear nuevos métodos que funcionen en estos contextos distribuidos y de ser posible que aprovechen la separación de los datos para realizar mejores análisis.

Problema de investigación

Los métodos existentes no han resuelto totalmente la selección de atributos con un costo computacional adecuado y hasta el momento el problema de la selección de rasgos en ambientes distribuidos no ha sido abordado suficientemente. Esto afecta a los algoritmos de aprendizaje automatizado encargados de extraer el conocimiento inherente a los datos.

Objetivo general

Desarrollar un método de selección de rasgos combinando la Optimización basada en Colonias de Hormigas, la Teoría de Conjuntos Aproximados y la Teoría de la Información en problemas de aprendizaje supervisado y extenderlo para fuentes de datos distribuidas.

Objetivos específicos

1. Utilizar un modelo que combina la Optimización basada en colonias de hormigas, conceptos de la Teoría de los Conjuntos Aproximados y la medida Ganancia de información de la Teoría de la Información como método de selección de rasgos para fuentes de datos distribuidas.
2. Establecer el valor de la ganancia de información como metadato para la cooperación entre los subsistemas que colaboran.
3. Extender el modelo a una variante distribuida que le dé solución al problema de selección de rasgos para múltiples fuentes de datos.

Preguntas de la investigación

1. ¿Será eficaz el método realizando el intercambio de metainformación propuesto?
2. ¿Se obtendrán mejores resultados aplicando el método de selección de rasgos para ambientes distribuidos que si se aplican algoritmos de selección de rasgos locales a cada fuente de datos?

Hipótesis

El uso de la ganancia de información como medida heurística en los métodos de selección de rasgos combinando la optimización basada en colonias de hormigas y la Teoría de Conjuntos

Aproximados en un contexto distribuido, permite construir competentes algoritmos de selección de rasgos.

Justificación

La falta de estudios en el área de la selección de rasgos en ambientes distribuidos hace que esta investigación contribuya a incrementar los conocimientos en el campo. Permite mostrar además, que la cooperación mediante el intercambio de metadatos entre subsistemas distribuidos puede ayudar a encontrar mejores soluciones.

1 La selección de rasgos y sus componentes

1.1 Descubrimiento de conocimiento, Minería de Datos y Minería de Datos Distribuida.

Se estima que la cantidad de información en el mundo se duplica cada 20 meses. De igual manera, las herramientas usadas en los diferentes campos del conocimiento deben desarrollarse para combatir ese abrumador crecimiento. Tradicionalmente, los datos eran analizados de forma manual, mediante métodos matemáticos y estadísticos. A medida que los datos crecen y superan la capacidad manual humana, se va haciendo imposible este tipo de análisis, debido al tiempo que requiere, al alto costo y a la alta susceptibilidad a errores.

Es por este motivo que surge la necesidad de un proceso de búsqueda de conocimiento en los datos de forma automática. Así surge el proceso conocido como *Descubrimiento de conocimiento en bases de datos* (del inglés Knowledge Discovery in Databases, KDD) (Frawley et al., 1992, Jensen, 2005). Este proceso consiste en la extracción no trivial de información potencialmente útil, mediante la identificación de patrones y relaciones existentes en los datos. Según Jensen, KDD puede descomponerse en los siguientes pasos:

1. Selección de los datos.

   Se realiza una selección de los datos que se utilizarán para la extracción de conocimiento.
2. Limpieza y pre procesamiento.

   En esta fase se preparan los datos con el fin de mejorar la calidad de la información contenida en ellos mediante la reducción de ruido, discretización de variables en caso de ser necesaria, resolución de valores ausentes.
3. Reducción de los datos.

   La mayoría de los conjuntos de datos poseen un alto grado de redundancia y características o rasgos que no aportan información ninguna y que pueden afectar el rendimiento e incluso la eficacia de la fase siguiente. Es en esta fase en la que se seleccionan los atributos más significativos, o sea, los que mejor representan la

información contenida en los datos. Es esta fase el centro de interés del presente trabajo.

4. Minería de Datos.

   Como parte esencial del descubrimiento de conocimiento se encuentra la *Minería de Datos* (del inglés Data Mining, DM). Esta se define como el proceso automático o semiautomático de reconocimiento de patrones significativos en los datos, de modo que se obtenga alguna ventaja en el uso de estos (Witten and Frank, 2005).

5. Interpretación y evaluación de los resultados.

   Se realiza una validación de los datos con el fin de determinar su utilidad, novedad y veracidad.

Es muy frecuente encontrar solapados la Minería de Datos con el Aprendizaje Automático y la Estadística, pues sus dominios de aplicación tienen mucho en común. Cabe destacar que la estadística fue el primer campo de la ciencia en ser usado para extraer información útil de los datos, y junto a otras disciplinas, forma la base de la Minería de Datos. El aprendizaje automático tiene como objetivo hacer que las computadoras aprendan. Este aprendizaje puede definirse como tener o adquirir conocimiento de algo y poseer la habilidad para usar ese conocimiento (Witten and Frank, 2005), dígase por ejemplo, predecir un evento desconocido sabiendo cómo han ocurrido eventos anteriores. Otras definiciones establecen como programa que aprende, aquel que mejora su desempeño a través del tiempo y del entrenamiento (Mitchell, 1997, Witten and Frank, 2005). Dado que la Minería de Datos trata acerca de encontrar y describir patrones, contando con una descripción explícita de su estructura, estos pueden considerarse como el conocimiento adquirido y ser usados para mejorar el desempeño de un sistema que los requiera. De ahí, que la Minería de Datos pueda usarse para aprender.

Contexto distribuido

Cuando los datos se encuentran separados geográficamente, ya sea en fuentes distribuidas o en múltiples unidades de cómputo, su análisis requiere el uso de alguna tecnología de Minería de Datos diseñada especialmente para estos ambientes distribuidos. Esto se debe a que en la mayoría de los casos es poco factible, y en ocasiones imposible, la unión de los datos en un solo

volumen, debido a restricciones tecnológicas, económicas o legales. El campo de la Minería de Datos Distribuida (del inglés Distributed Data Mining, DDM) trata esta problemática, cada vez más importante en nuestra sociedad del conocimiento.

Un sistema de Minería de Datos distribuida está compuesto por varios subsistemas, donde cada uno tiene acceso solo a un volumen de datos y no a los restantes volúmenes. Un principio elemental es que tal técnica debe ser capaz de aprender de un ambiente de datos distribuido sin comprometer la privacidad de los datos de cada subsistema. Para lograr dicho aprendizaje es necesario encontrar una forma de cooperación entre los elementos del sistema de manera que se cumpla esa regla básica, por ejemplo, intercambiando entre los subsistemas cierta meta información o meta datos (información acerca de los datos).

1.2 Selección de rasgos

La alta dimensionalidad presente en la mayoría de los datos, fundamentalmente en los extraídos de fenómenos poco conocidos de la naturaleza hace que los métodos de extracción de conocimiento sean poco efectivos, si son aplicados a los datos en bruto (Zhang et al., 2003, Zhong et al., 2001). Tradicionalmente, cuando quiere obtener datos, el hombre trata de obtener la mayor cantidad posible con la mayor cantidad de variables, con el fin de no perder información y confiando en su poder discriminatorio para luego tomar las que más información presentan. Sin embargo, cuando la cantidad de datos supera la capacidad de análisis e interpretación directa del hombre se hace extremadamente difícil determinar cuáles son los rasgos más importantes, principalmente cuando estos representan algún proceso poco estudiado y conocido por el hombre; y lo que en un principio se hizo con la intención de obtener mucho conocimiento se convierte en una fuente de desinformación y oculta aún más el conocimiento del que los datos son portadores.

Por esta razón se hace necesaria una herramienta para automatizar el proceso de selección de rasgos y hacerlo efectivo. La selección de rasgos tiene como meta reducir la dimensionalidad de los datos y seleccionar el mejor subconjunto de atributos, basándose en algún criterio de clasificación (Liu et al., 2010, Liu and Motoda, 2007). Esta selección se logra mediante la

eliminación de rasgos redundantes e irrelevantes (Bell and Wang, 2000, Blum and Langley, 1997), que alcanzan una mejor generalización del algoritmo de aprendizaje.

En las tareas de selección de rasgos, los datos generalmente contienen muchos atributos, lo cual conduce a una enorme cantidad de subconjuntos de atributos de los cuales se puede hacer la selección; nótese que para m rasgos la cantidad total de subconjuntos es de $2^m - 1$, haciendo que este problema sea NP-complejo (Hu et al., 2007). Como el objetivo es encontrar un subconjunto de atributos que maximice (minimice) cierta función o criterio, la selección de rasgos puede verse como un problema de búsqueda (Langley, 1994, Siedlecki and Sklansky, 1988) en un espacio formado por todos los subconjuntos posibles. Dado que el tamaño del espacio de búsqueda es exponencial con respecto a la cantidad de rasgos, es evidente que una búsqueda exhaustiva no es viable para dar solución a este problema (Hu et al., 2007), incluso para conjunto de datos de mediano tamaño (Blum and Langley, 1997). Por tal motivo, es necesaria la utilización de estrategias de búsqueda heurística o aleatoria con el objetivo de reducir la porción del espacio sobre la que se busca, a pesar de que de esta manera no se garantice encontrar el óptimo, a diferencia de la búsqueda exhaustiva.

Los procedimientos de selección de rasgos constan de dos componentes principales: una función de evaluación y un método de generación de subconjuntos o método de búsqueda (Ruiz et al., 2008). La primera es una función basada en un criterio determinado que permite decidir cuán *bueno* es un subconjunto, esto es, cuánta información o conocimiento preserva de los datos originales. El método de generación de subconjuntos no es más que la estrategia de búsqueda que permite moverse por el espacio de búsqueda y seleccionar los posibles subconjuntos de atributos.

Atendiendo a cómo y cuándo la calidad de los subconjuntos de rasgos es evaluada, los métodos de selección de rasgos se clasifican en tres categorías: filtros, envolventes (del inglés wrapper) y los empotrados (built-in)(Liu et al., 2010). La primera categoría, como su nombre lo indica, funciona como un filtro de los atributo y es un proceso independiente de la fase de aprendizaje. En los envolventes, la selección de rasgos y el proceso de inducción son procesos codependientes, donde la selección hace uso del proceso de aprendizaje para evaluar la calidad

de los subconjuntos de atributos seleccionados. Los métodos empotrados, realizan la selección de atributos durante el proceso de entrenamiento del algoritmo de aprendizaje, es decir, el algoritmo de aprendizaje tiene incluida su propia forma de seleccionar los rasgos más significativos. Como ejemplo de algoritmos con selección empotrada podemos encontrar los árboles de decisión, los cuales utilizan sólo los atributos necesarios para obtener una descripción consistente con el conjunto de aprendizaje.

Algunos de los métodos existentes para la selección de rasgos pueden encontrarse en: (Narendra and Fukunaga, 1977, Sheinvald et al., 1990, Almuallim and Dietterich, 1992, Almuallim and Dietterich, 1994, Liu et al., 1998, Kira and Rendell, 1992, Kononenko, 1994, Pudil et al., 1994, Doak, 1992, Yu and Liu, 2004, Battiti, 1994, Hall, 2000, Mucciarde and Gose, 1971, K.Wang and Sundaresh, 1998, Liu and Setiono, 1996, Balamurugan and Rajaram, 2009, Gadat and Younes, 2007). Debido a limitaciones de los métodos existentes, se ha mostrado una tendencia al uso de métodos de búsqueda basados en la Inteligencia Colectiva (del inglés, Swarm Intelligence). Estos métodos tienen la ventaja de encontrar una mayor cantidad de subconjuntos de atributos, lo cual despierta un interés especial en diversos dominios. Algunos de estos métodos que utilizan Optimización basada en enjambres de partículas y Optimización basada en colonias de hormigas pueden encontrarse en: (Firpi and Goodman, 2004, Al-Ani, 2005, Jensen and Shen, 2003, Jensen and Shen, 2005, Gómez Díaz, 2011).

Una consecuencia directa del uso de métodos heurísticos para la búsqueda de subconjuntos óptimos de atributos es el hecho de que no existe un método que garantice ser mejor que cualquier otro, como plantean Wolpert y Macready en el teorema *No Free Lunch* (Wolpert and Macready, 1997). Por este motivo, no es extraña la aparición constante de nuevos métodos para resolver este problema, lo que justifica parte del presente trabajo.

Reducto

Una noción muy asociada al problema de selección de rasgos es el concepto de reducto, de hecho, el término *reducto* corresponde a una amplia clase de conceptos (Pawlak and Skowron, 2007); sin embargo, algo común entre todos ellos es que son usados para eliminar atributos redundantes. Una definición muy general del término consiste en un subconjunto de atributos

tal que, como conjunto, este es suficiente para expresar la información contenida en los datos originales y donde todos sus elementos son individualmente necesarios. Dicho de otro modo, conociendo un reducto es posible eliminar de los datos todos los atributos no contenidos en este sin sufrir ninguna pérdida de información, quedando como resultado un subconjunto del cual no es posible eliminar un atributo y a la vez conserve su condición de reducto.

1.3 Teoría de Conjuntos Aproximados

En las últimas dos décadas, la Teoría de Conjuntos Aproximados ha ganado un gran interés en la comunidad científica gracias a su capacidad para extraer dependencias entre los datos y reducir la dimensionalidad de estos sin requerir información adicional y sin afectar su semántica. La Teoría de Conjuntos Aproximados fue introducida por Z. Pawlak en 1982 (Pawlak, 1982, Pawlak, 1991) y permite encontrar en un conjunto de datos, el subconjunto de atributos que mejor define a los datos, es decir, el subconjunto de rasgos con que se puede representar los datos con la mínima pérdida de información. Visto desde el contexto de clasificación supervisada, estos son los rasgos que mejor predicen el atributo de decisión.

Algunas de las ventajas del uso de la Teoría de Conjuntos Aproximados para la selección de rasgos son las siguientes:

- Posee la capacidad de realizar la reducción de la cantidad de atributos sin destruir ni alterar la información acumulada en los datos.
- Como entrada sólo necesita el conjunto de datos sin ningún tipo de información adicional (Düntsch and Gediga, 1998); a diferencia de los conjuntos difusos y muchos de los métodos estadísticos en los que hay que introducir determinados parámetros al método, tales como rango de valores, umbrales de ruido, o se requiere un conocimiento previo de la naturaleza de los datos.
- Es posible aplicar esta teoría a conjuntos de datos que presentan inconsistencias. Una inconsistencia consiste en la existencia de dos objetos o casos con iguales valores en los atributos de predictores y valores diferentes en el atributo de decisión (Magnani, 2003).

Definiciones principales de la Teoría de Conjuntos Aproximados

Sistema de información y sistema de decisión

La forma más sencilla de representación de los datos para su análisis con conjuntos aproximados es mediante una tabla en la que cada fila representa un objeto, un caso, un paciente o un evento y cada columna representa un atributo, o sea, un rasgo, una característica, una propiedad presente en cada uno de los objetos. Este tipo de tablas recibe el nombre de *sistema de información*. Cuando a dicho sistema se le agrega un atributo que indica una categoría, clase o grupo al que pertenece cada objeto, entonces se obtiene un *sistema de decisión*. Definiciones más formales son las siguientes:

Definición 1-1 Sistema de información

Se denomina sistema de información al par (U, A) donde U es el conjunto finito no vacío de objetos llamado el universo y A es el conjunto finito no vacío de atributos tal que $\exists f \forall_{(u,a)\in U\times A}\ f: U \times A \to V$ y f es una función llamada función de información (Komorowski et al., 1999). El conjunto V es el conjunto de los valores presentes en la tabla.

Definición 1-2 Sistema de decisión

Se denomina sistema de decisión o tabla de decisión al sistema de información formado por el par $(U, A \cup \{d\})$ donde $d \notin A$ es llamado atributo de decisión y el conjunto A es llamado atributos condicionales o condiciones.

Relación de inseparabilidad y clases equivalencia

Definición 1-3 Relación de inseparabilidad

Sea $S = (U, A)$ un sistema de información, entonces:

$$\forall_{B\subseteq A}\exists_{IND(B)}\ IND(B) = \{(x, y) \in U^2 |\ \forall_{b\in B}\ f(x, b) = f(y, b)\}$$

A $IND(B)$ se denomina *relación de B-inseparabilidad* (Komorowski et al., 1999)(del inglés, B-indiscernibility relation), y consiste, expresado en lenguaje natural, en la relación de equivalencia inducida por el conjunto B de atributos, es decir, si el par $(x, y) \in IND(B)$

entonces los objetos x e y son imposibles de diferenciar basándose en los atributos contenidos en B. Nótese que esta es una relación de equivalencia y por lo tanto produce una partición del universo de objetos.

Las clase de equivalencia de un objeto x del universo U inducida por el conjunto de atributos B se denota $[x]_B$ y es el conjunto de todos los objetos indiscernibles de x. Esto es:

$$[x]_B = \{y|(x,y) \in IND(B)\}$$

Aproximaciones de los conjuntos

En muchos sistemas de decisión obtenidos a partir de problemas reales no es posible definir el atributo de decisión a partir de la partición originada por la relación de inseparabilidad. Normalmente se esperaría que objetos pertenecientes a una misma clase de equivalencia posean el mismo valor en el atributo de decisión, sin embargo, esto no siempre ocurre en los problemas del mundo real, pues en muchas ocasiones no se mide el atributo o conjunto de atributos capaces de discernir objetos con distinta clasificación, o simplemente puede alterarse el criterio de clasificación en determinado momento de la obtención de los datos (Magnani, 2003). Es por este motivo que surge la necesidad de emplear los conjuntos aproximados. De esta manera se puede determinar cuáles objetos positivamente pertenecen a cierta categoría, cuáles no pertenecen a ella y cuáles son los que yacen en la frontera o los que pueden o no pertenecer a dicha clasificación.

Para las siguientes definiciones se usarán las siguientes declaraciones. Sea $S = (U, A)$ un sistema de información y sean $B \subseteq A$ y $X \subseteq U$

Definición 1-4 Aproximación inferior

La aproximación inferior de X inducida por B o aproximación B-inferior se denota $\underline{B}(X)$ y se define como:

$$\underline{B}(X) = \{x|[x]_B \subseteq X\}$$

Definición 1-5 Aproximación superior

La aproximación superior de X inducida por B o aproximación B-superior se denota $\overline{B}(X)$ y se define como:

$$\overline{B}(X) = \{x | [x]_B \cap X \neq \emptyset\}$$

Los objetos pertenecientes a $\underline{B}(X)$ pueden ser con toda certeza clasificados como miembros de X basándose en el conocimiento proporcionado por B, mientras que los objetos pertenecientes a $\overline{B}(X)$ sólo pueden ser clasificados como posibles miembros de X.

Definición 1-6 Región frontera

El conjunto denotado por $BN_B(X)$ es denominado región frontera o región límite inducida por B y se define por:

$$BN_B(X) = \overline{B}(X) - \underline{B}(X)$$

Los elementos pertenecientes a este conjunto no puede asegurarse que puedan ser clasificados como miembros de X con la base del conocimiento aportado por B.

Dependencias entre atributos

Una importante materia en el análisis de los datos es el descubrimiento de dependencias entre los atributos. Intuitivamente un conjunto de atributos D depende totalmente de un conjunto de atributos C, denotado $C \Rightarrow D$ si todos los valores de los atributos de D pueden ser determinados por valores de atributos de C. La dependencia entre atributos puede ser definida con más precisión de la siguiente manera.

Definición 1-7 Grado de dependencia

Sean $C, D \subseteq A$, se dice que D depende de C en un grado $k \in [0,1]$ denotado $C \Rightarrow_k D$ cuando:

$$k = \gamma_C(D) = \frac{\sum_{X \in U/_D} |\underline{C}(X)|}{|U|}$$

Siendo $U/_D$ la partición de U generada por D. Si $k = 1$ se dice que D depende totalmente[1] de C, si $0 < k < 1$ entonces D depende parcialmente de C con grado de dependencia k. Cuando la dependencia es total puede verse que $IND(C) \subseteq IND(D)$, esto significa que la partición generada por C es más refinada que la generada por D. En términos de lenguaje natural, $\gamma_C(D)$ expresa la proporción de objetos del universo que pueden ser correctamente clasificados en bloques de la partición $U/_D$ empleando C.

En el contexto de aprendizaje supervisado cuando la tabla de datos es un sistema de decisión, al grado de dependencia entre un conjunto de atributos condicionales y el atributo de decisión se le llama *calidad de aproximación de la clasificación*, y esta es una medida que expresa la proporción de objetos que con certeza están correctamente clasificados en el sistema.

Definición 1-8 Calidad de aproximación de la clasificación

Sean $S = (U, A \cup \{d\})$ un sistema de decisión y $B \subseteq A$, se denomina calidad de aproximación de la clasificación a $\gamma_B(d)$. Si $\gamma_A(d) = 1$ entonces el sistema S es consistente, es decir, no presenta inconsistencias.

Utilizando la medida calidad de aproximación de la clasificación se puede definir el término reducto como un subconjunto de atributos $R \subseteq A$ tal que $\gamma_R(d) = \gamma_A(d)$ y $\forall_{B \subset R}\, \gamma_B(d) \neq \gamma_A(d)$. Esto significa que un reducto es cualquier subconjunto de atributos condicionales tales que la calidad de aproximación de la clasificación sea la misma que para el conjunto de todos los atributos condicionales, y a su vez, todo reducto pierde dicha condición si se le sustrae alguno de sus atributos. El uso de reductos en la selección y reducción de atributos ha sido ampliamente estudiado (Kohavi and Frasca, 1994, Lazo et al., 2001, Bello et al., 2005, Caballero and Bello, 2006).

[1] Este es el caso de dependencia funcional estudiada en Bases de datos.

1.4 Teoría de la Información

La Teoría de la Información es una rama de la matemática introducida por Claude Elwood Shannon en 1948 (Shannon, 1948). Esta se encarga del estudio de la información en términos estrictamente estadísticos, bajo el supuesto de que puede ser tratada de manera semejante a como son tratadas otras magnitudes físicas como la masa y la energía. La Teoría de la Información permite, al igual que la Teoría de Conjuntos Aproximados, realizar el análisis de los datos sin alterarlos y sin requerir información adicional; además de ser capaz de extraer información de la estructura de los datos y cuantificar la cantidad de información que estos presentan, lo que le ha servido para ganar una gran popularidad en la comunidad científica.

Entropía

El término *entropía* (de origen griego *tropos*, que significa cambio, transformación) surgió como una variable de estado en la termodinámica clásica, siendo Rudolph Clausius (Clausius, 1865) quien primero desarrolló explícitamente la idea como parte de lo que después sería la segunda ley de la termodinámica. Luego Ludwig Boltzmann, Josiah Willard Gibbs y James Clerk Maxwell extendieron la idea a la mecánica estadística a fines del siglo XIX, formando de esta manera un nuevo punto de vista de la entropía fuera de la termodinámica clásica y la física. Sin embargo, fue Shannon quien por primera vez logró definir la entropía desde el enfoque de la Teoría de la Información y quién demostró que su definición es la única función capaz de medir la incertidumbre de una lista de eventos con probabilidades de ocurrencia conocidas y que cumpla ciertas propiedades predefinidas (Shannon, 1948). Debido a la gran cantidad de definiciones diferentes de entropía, los dominios en los que es utilizada y los diferentes significados que se le atribuyen, la definición en la que se enmarca el presente trabajo es conocida mundialmente como *la entropía de Shannon*.

Definición 1-9 Entropía

Sea X una variable aleatoria, $V = \{v_1, v_2, \dots, v_n\}$ el conjunto de los valores que puede tomar X y $p_i = p(v_i)$ la probabilidad de que la variable tome el valor v_i; se define por entropía asociada a X a la función (Shannon, 1948):

$$H(X) = -K \sum_{i=1}^{n} p_i \log_a p_i$$

Donde K es una constante positiva usada para definir las unidades en las que se mide la información o la incertidumbre. Haciendo $K = \frac{1}{\log_a 2}$ la definición puede simplificarse a:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

De esta forma, la unidad con la que se mide la información es el dígito binario o *bit*. Esta medida puede verse como la cantidad de incertidumbre involucrada en la variable X. Nótese que si todas las probabilidades p_i son iguales a $\frac{1}{n}$ la entropía alcanza el máximo valor igual a $\log_2 n$, siendo esta la configuración de probabilidades con la mayor incertidumbre, ya que es la que más dificulta la predicción de qué valor tomará la variable. Sin embargo, el mínimo valor de entropía $H(X) = 0$ se alcanza cuando todas las probabilidades son cero, excepto una con valor igual a la unidad, esto se explica por el hecho de que en este caso no hay incertidumbre, pues con toda certeza se puede saber el valor que tomará la variable.

La utilidad de esta medida en el contexto de la Minería de Datos y la selección de rasgos radica en que puede ser aplicada a un conjunto de datos para conocer la información aportada por uno o varios de sus atributos. Así se puede definir la entropía para ser usada en un sistema de información $S = (U, A)$, siendo $B \subseteq A$:

$$H(B) = -\sum_{X \in U/B} \frac{|X|}{|U|} \log_2 \frac{|X|}{|U|}$$

Donde U/B es la partición del universo U generada por el subconjunto de atributos B.

Para el caso particular en que $S = (U, A \cup \{d\})$ es un sistema de decisión, al valor de la entropía para el atributo de decisión d se le llama *entropía del sistema* y se denomina $H(D)$ donde $D = \{d\}$. Este es un indicador de la distribución de las clases en la tabla. Si $H(D) = 0$ significa que todos los objetos pertenecen a la misma clase; si $H(D)$ es máximo, entonces existe la misma cantidad de objetos en casa clase.

Shannon también define la entropía condicional para determinar la incertidumbre involucrada en la sucesión de dos eventos aleatorios no necesariamente independientes.

Definición 1-10 Entropía condicional

Sean X y Y dos variables aleatorias no necesariamente independientes, $\{x_1, x_2, \dots, x_m\}$ y $\{y_1, y_2, \dots, y_n\}$ los valores posibles de X y Y respectivamente, se define como entropía condicional de Y dado X, como el promedio de la entropía de Y para cada valor de X, pesándolo de acuerdo a la probabilidad de obtener ese valor de X, esto es dado por:

$$H_X(Y) = -\sum_{i,j} p(x_i, y_j) \log_2 p(y_j | x_i)$$

Sustituyendo el valor de la probabilidad condicional:

$$H_X(Y) = -\sum_{i,j} p(x_i, y_j) \log_2 p(x_i, y_j) + \sum_{i} p(x_i) \log_2 p(x_i)$$

Esto es:

$$H_X(Y) = H(X, Y) - H(X)$$

Para un sistema de decisión, el valor de la entropía condicional del atributo de decisión para un subconjunto de atributos condicionales da una medida de la incertidumbre en la clasificación, conocidos los valores de estos atributos. Sea $S = (U, A \cup \{d\})$ el sistema de decisión utilizado anteriormente, $B \subseteq A$ y $D = \{d\}$ el conjunto unitario formado por el atributo de decisión, la medida entropía condicional o relativa del sistema conocidos los valores de B se define por:

$$H_B(D) = H(B, D) - H(B) = -\sum_{X \in U/_{D \cup B}} \frac{|X|}{|U|} \log_2 \frac{|X|}{|U|} + \sum_{Y \in U/_{B}} \frac{|Y|}{|U|} \log_2 \frac{|Y|}{|U|}$$

Usando la entropía del sistema y la entropía condicional se puede medir la cantidad de información que aporta un subconjunto de atributos a un sistema de decisión. Con este objetivo surge la medida *ganancia de información*.

Definición 1-11 Ganancia de información

Sean $S = (U, A \cup \{d\})$ sistema de decisión, con d como atributo de decisión, $B \subseteq A$ y $D = \{d\}$ se define por ganancia de información:

$$IG_B(D) = H(D) - H_B(D)$$

Se entiende por ganancia de información a la entropía del sistema menos la entropía condicional conocido el valor de los atributos en B, o sea, es una medida de cuánto decrece la incertidumbre del sistema cuando se conoce el valor de un conjunto de atributos, y por lo tanto, mientras menor sea la incertidumbre, mayor es la posibilidad de clasificar correctamente los objetos. Por lo tanto, mientras mayor sea la ganancia de información para un subconjunto de atributos, mayor información aporta este al sistema.

Existen en la literatura una gran cantidad de algoritmos de clasificación, de selección de rasgos y de otra índole que usan estas medidas de la Teoría de la Información, entre ellos (Quinlan, 1986, Shie and Chen, 2007, Díaz Galiano et al., 2007, Choo et al., 2008, Jensen and Shen, 2003) por citar algunos.

1.5 Optimización basada en colonias de hormigas

La Optimización basada en colonias de hormigas (ACO, de *Ant Colony Optimization*) (Dorigo and Stutzle, 2004, Dorigo et al., 2006) pertenece a un paradigma inteligente llamado *Inteligencia Colectiva* para la solución de problemas de optimización inspirado en los comportamientos de diversas criaturas de la naturaleza. El caso de ACO basa su funcionamiento en la conducta de la las hormigas reales en la búsqueda de alimentos para resolver problemas de optimización discretos. Esta es una metaheurística poblacional que realiza un proceso constructivo y estocástico guiado por rastros de feromona[2] depositados por cada hormiga, lo que da una medida de cuán deseable es un camino recorrido a través de una función de visibilidad que evalúa la calidad de cada paso durante el desplazamiento. Es un ejemplo clásico de comunicación indirecta que ocurre cuando un individuo altera el medio en que se desarrolla y

[2] Sustancia química olorosa que depositan las hormigas en su recorrido. La intensidad de esta sustancia disminuye a través de un proceso de evaporación.

los otros son capaces de captar estos cambios siguiendo así la idea original sobre la que están basados los algoritmos de inteligencia de enjambre.

Los algoritmos basados en colonias de hormigas son procesos iterativos en los que en cada iteración se sitúa una colonia de hormigas y cada una de ellas construye una posible solución al problema. Estos métodos son esencialmente constructivos, es decir, en cada iteración cada hormiga construye una posible solución recorriendo un grafo. De forma general, en un algoritmo ACO, los arcos del grafo por los que se mueven las hormigas poseen dos tipos de información que guían su movimiento:

- *Información heurística*, mide la preferencia heurística de moverse desde el nodo i hasta el nodo j; es decir, la preferencia a recorrer la arista $a_{i,j}$. Se denota por $\eta_{i,j}$. Las hormigas no modifican esta información durante la ejecución del algoritmo.
- *Información de los rastros artificiales de feromona*, mide la *deseabilidad aprendida* del movimiento del nodo i al nodo j. Imita de forma numérica a la feromona real que depositan las hormigas naturales. Esta información se modifica durante la ejecución del algoritmo dependiendo de las soluciones encontradas por las hormigas. Se denota por $\tau_{i,j}$.

El modo de operación de un algoritmo ACO (Dorigo, 1992, Dorigo and Stutzle, 2004) es el siguiente: las m hormigas (artificiales) de la colonia se mueven, concurrentemente y de manera asíncrona, a través de los estados adyacentes del problema (que puede representarse en forma de grafo con ponderaciones o sin ellas). Este movimiento se realiza siguiendo una regla de transición que está basada en la información local disponible en las componentes (nodos). Esta información local incluye la información heurística y memorística (rastros de feromona) para guiar la búsqueda. Las hormigas construyen incrementalmente soluciones al moverse por el grafo de construcción. Opcionalmente, las hormigas pueden depositar feromona cada vez que crucen un arco (conexión) mientras que construyen la solución (*actualización en línea paso a paso de los rastros de feromona*). Una vez que cada hormiga ha generado una solución, ésta se evalúa y el agente puede depositar una cantidad de feromona en dependencia de la calidad de su solución (*actualización en línea de los rastros de feromona*). Esta información guiará la

búsqueda de las otras hormigas de la colonia en el futuro. Además, el modo de operación genérico de un algoritmo ACO incluye dos procedimientos adicionales, la evaporación de los rastros de feromona y las acciones del proceso demonio[3]. La evaporación de feromona la lleva a cabo el entorno, se usa como un mecanismo que evita el estancamiento en la búsqueda y permite que las hormigas busquen y exploren nuevas regiones del espacio. Las acciones del proceso demonio constituyen una funcionalidad opcional (que no tiene un contrapunto natural) para implementar tareas desde una perspectiva global que no pueden llevar a cabo las hormigas por la perspectiva local que ofrecen. Ejemplos son: observar la calidad de todas las soluciones generadas y depositar una nueva cantidad de feromona adicional sólo en las componentes asociadas a algunas soluciones, o aplicar un procedimiento de búsqueda local a las soluciones generadas por las hormigas antes de actualizar los rastros de feromona. En ambos casos el demonio reemplaza la actualización en línea a posteriori de feromona y el proceso pasa a llamarse *actualización fuera de línea (offline) de rastros de feromona*.

El significado que se le da a los rastros de feromona y a la función heurística en ACO es algo que depende totalmente del problema para resolver. Por ejemplo, cuando se está en presencia de un problema de secuenciación como el problema del viajero vendedor (Dorigo and Gambardella, 1997) o de asignación cuadrática (Gambardella et al., 1999) donde el orden en que aparecen las componentes en una solución influye en el valor de esta, la feromona es depositada en los arcos del grafo, para así diferenciar las distintas secuencias. Sin embargo, en problemas donde el valor de las soluciones depende solamente de sus componentes sin importar el orden en que aparecen, como los problemas de selección de rasgos (Bello et al., 2005), partición de conjuntos (Crawford and Castro, 2006), la feromona es asociada a los nodos del grafo.

De forma general, el funcionamiento de un algoritmo ACO (Dorigo and Stutzle, 2004) es el siguiente:

[3] Término de UNIX para referirse a un proceso que se ejecuta en el *fondo* y se activa solamente en determinadas ocasiones.

```
Procedimiento metaheurística ACO;
   Actividades Programadas
       Construir Soluciones de las Hormigas
       Actualizar Feromonas
       Evaporación de la Feromona
       Acciones del proceso demonio (opcional)
   Fin de las Actividades Programadas
Fin del Procedimiento
```

Figura 1 Procedimiento general de ACO

Este procedimiento se anida en el siguiente procedimiento iterativo:

```
Paso1: Inicializar los valores de feromona
           Iteración Actual=1
Paso2: Repetir
           Procedimiento  metaheurística ACO
           Iteración Actual = Iteración Actual +1
       Hasta que: criterio de parada
```

Figura 2 Estructura genérica de ACO

Para los métodos de ACO existen distintos criterios de parada (Dorigo and Stutzle, 2003), en esta investigación el criterio de parada se cumple cuando se alcanza una cantidad máxima de iteraciones.

Observando las aplicaciones actuales de ACO, se pueden identificar algunas directivas sobre cómo solucionar problemas utilizando esta metaheurística (Dorigo et al., 2006). Estas directivas se pueden resumir en las seis tareas de diseño que se enumeran a continuación:

1. Representar el problema como un conjunto de componentes (nodos) y transiciones (aristas) a través de un grafo que será recorrido por las hormigas para construir soluciones.

2. Definir de manera apropiada en base a las características del problema, el significado de los rastros de feromona τ.
3. Definir de manera apropiada la preferencia heurística o función de visibilidad η asociada a cada componente o transición.
4. Si es posible, implementar una búsqueda local eficiente para mejorar las soluciones obtenidas por ACO.
5. Escoger un algoritmo de ACO específico y aplicarlo al problema que hay que solucionar teniendo en cuenta las características propias de cada uno de estos algoritmos.
6. Refinar los parámetros del algoritmo de ACO seleccionado.

Dentro de los algoritmos de ACO las diferencias fundamentales radican en la regla de transición que utilizan para la construcción de las soluciones y en el tratamiento que le dan a los rastros de feromona. Debido a esto, aparecen en la literatura distintos algoritmos ACO.

Diversas variantes de la familia de algoritmos basados en colonias de hormigas

Entre los algoritmos ACO disponibles para problemas de optimización combinatoria (Dorigo and Blum, 2005) se encuentran: el Sistema de Hormigas (*Ant System*, AS) (Dorigo et al., 1996), el Sistema de Colonia de Hormigas (*Ant Colony System*, ACS) (Dorigo and Gambardella, 1997), el Sistema de Hormigas Máximo-Mínimo (*Max-Min Ant System*, MMAS) (Stützle and Hoos, 2000), entre otros.

A continuación se presenta una descripción general de los algoritmos Sistema de hormigas y Sistema de colonia de hormigas debido a que fueron los estudiados para llevar a cabo este trabajo.

Sistema de hormigas

El Sistema de Hormigas, desarrollado por Dorigo en su tesis doctoral en 1992 (Dorigo, 1992), fue el primer algoritmo de ACO. Su versión actual (*Ant Cycle*) apareció conjuntamente con otras variantes de este, como el Sistema de Hormigas Densidad (*Ant Density*) y Sistema de Hormigas Cantidad (*Ant Quantity*). El AS se caracteriza por el hecho de que, la actualización de feromona

se realiza una vez que todas las hormigas hayan completado sus soluciones, y se lleva a cabo como sigue: todos los rastros de feromona se reducen en un factor constante, implementándose de esta manera la evaporación de feromona según la ecuación:

$$\tau_{i,j} \leftarrow (1-\rho)\tau_{i,j} \quad \rho \in (0,1]$$

Donde ρ se conoce como constante de evaporación y es la encargada de reducir los rastros de feromona para evitar el estancamiento de las soluciones y $\tau_{i,j}$ representa la cantidad de feromona asociada al arco (i,j).

A continuación, cada hormiga de la colonia deposita una cantidad de feromona en función de la calidad de su solución, según la ecuación:

$$\tau_{i,j} \leftarrow \tau_{i,j} + \Delta\tau^k \quad \forall_{(i,j)\in S^k}$$

Donde $\Delta\tau^k = f\Big(C(S^k)\Big)$, representa la cantidad de feromona a depositar por la hormiga k en cada arco (i,j) de su solución encontrada S^k. Este valor depende de la calidad de la solución $C(S^k)$.

Las soluciones en el AS se construyen como sigue. En cada paso de construcción, una hormiga k escoge ir al siguiente nodo con una probabilidad que se calcula como:

$$p_{i,j}^k = \frac{(\tau_{i,j})^\alpha (\eta_{i,j})^\beta}{\sum_{l\in N_i^k}(\tau_{i,l})^\alpha (\eta_{i,l})^\beta} \quad \text{si } j \in N_i^k$$

Donde N_i^k es el vecindario alcanzable por la hormiga k cuando se encuentra en el nodo i. Los parámetros α y β controlan el proceso de búsqueda. Para $\alpha = 0$ se tiene una búsqueda heurística estocástica clásica, mientras que para $\beta = 0$ sólo el valor de la feromona tiene efecto. Un valor de $\alpha > 1$ conlleva a una rápida situación de convergencia (Dorigo et al., 1999). El valor $\tau_{i,j}$ representa el elemento la feromona en el arco (i,j) y $\eta_{i,j}$ se denomina función de visibilidad o función heurística y mide la calidad de un nodo j a partir del nodo i.

Sistema de colonia de hormigas

El Sistema de Colonia de Hormigas, es uno de los primeros sucesores del AS que introduce tres modificaciones importantes con respecto a dicho algoritmo ACO:

1. Utiliza una regla de transición distinta y más agresiva, denominada regla pseudoaleatoria proporcional. Sea k una hormiga situada en el nodo i, $q_0 \in [0,1]$ un parámetro y q un valor aleatorio en el mismo intervalo, el siguiente nodo j se elige como:

$$j = \max_{j \in N_i^k} \left\{ \left(\tau_{i,j}\right)^{\alpha} \left(\eta_{i,j}\right)^{\beta} \right\} \quad \text{si } q \leq q_0$$

   En caso contrario se utiliza la regla probabilística del AS con $\alpha = 1$.

   Como puede observarse, la regla tiene una doble intención: cuando $q \leq q_0$, utiliza en gran medida el conocimiento disponible (explotación), eligiendo la mejor opción con respecto a la información heurística y los rastros de feromona. Sin embargo, si $q > q_0$ se aplica una exploración controlada, tal como se hace en el AS.

2. Las hormigas aplican una actualización en *línea paso a paso* de los rastros de feromona que favorece la generación de soluciones distintas a las encontradas.

   Cada vez que una hormiga viaja por una arista (i, j), aplica la regla:

$$\tau_{i,j} \leftarrow (1 - \varphi)\tau_{i,j} + \varphi\tau_0$$

   Donde $\varphi \in (0,1]$ es un segundo parámetro de decremento de feromona y τ_0 es un valor constante. Como puede verse, la regla de actualización en *línea paso a paso* incluye tanto la evaporación de feromona como la actualización de la misma.

3. Se realiza una actualización *fuera de línea* de los rastros de feromona (acción del proceso demonio), donde el ACS sólo considera una hormiga concreta, la que generó la mejor solución global, $s_{\text{mejor-global}}$[4].

[4]Aunque en algunos trabajos iniciales se consideraba también una actualización basada en la mejor hormiga de la iteración, en ACS casi siempre se aplica la actualización basada en la mejor global.

La actualización de la feromona se lleva a cabo evaporando primero estos rastros en todas las conexiones utilizadas por la mejor hormiga global (es importante recalcar que, en el ACS, la evaporación de la feromona sólo se aplica a las conexiones de la solución, que es también la usada para depositar feromona) tal como sigue:

$$\tau_{i,j} \leftarrow (1-\rho)\tau_{i,j} \quad \forall_{(i,j)\in S_{\text{mejor-global}}}$$

A continuación se deposita feromona a los arcos que pertenecen a la mejor solución encontrada hasta el momento usando la regla:

$$\tau_{i,j} \leftarrow \tau_{i,j} + \rho\Delta\tau \quad \forall_{(i,j)\in S_{\text{mejor-global}}}$$

Donde $\Delta\tau = f\left(C\left(S_{\text{mejor-global}}\right)\right)$, es decir la cantidad de feromona está en dependencia de la calidad de la mejor solución encontrada hasta el momento $C\left(S_{\text{mejor-global}}\right)$.

En general, las soluciones obtenidas con la metaheurística ACO suelen ser de una calidad moderada. Esto se debe a que estos algoritmos se inclinan hacia una mayor exploración del espacio de búsqueda, por lo que es razonable aplicarle diversos algoritmos de búsqueda local (Crawford and Castro, 2006, Dorigo et al., 2006, Heinonen and Pettersson, 2007, Huang and Liao, 2008) o alguna estrategia para agregarle un mayor grado de explotación de las soluciones encontradas (Naimi and Taherinejad, 2009, Wong and See, 2009, Wu et al., 2009).

1.6 Consideraciones parciales

Por lo analizado a lo largo del presente capítulo se puede concluir que aún no existen métodos que resuelvan satisfactoriamente todas las necesidades del preprocesamiento de los datos para la extracción correcta y efectiva de conocimientos.

La revisión de los algoritmos de construcción de reductos muestra que la mayoría depende de estrategias heurísticas de búsqueda, lo que conlleva la continua aparición de nuevos métodos y la combinación de otros ya existentes para dar solución al problema presentado; pues, como ya se ha demostrado, no existe un modelo capaz de solucionarlo siempre con mejores resultados que los demás.

Debido a las características de los algoritmos ACO, estos son recomendables para solucionar problemas de selección de rasgos, pues su habilidad para encontrar soluciones en espacios de búsqueda tan grandes mediante la inteligencia de enjambres representa una ventaja muy apreciada cuando la posibilidad de aplicar modelos de búsqueda exhaustivos no son factibles.

Otros métodos dependen de información adicional suministrada conjuntamente con los datos, como el caso de umbrales de ruido o la longitud de los reductos para encontrar; lo que trae consigo una dependencia del criterio del usuario y una posible deficiente calidad en las soluciones encontradas. El uso de la Teoría de Conjuntos Aproximados y la Teoría de la Información ofrece la posibilidad de analizar los datos sin presentar ninguno de estos inconvenientes, lo que podría ser muy efectivo al formar parte de un modelo nuevo que permita la extracción de rasgos relevantes en un sistema o sistemas de datos.

2 Solución al problema de selección de rasgos

El uso de la Inteligencia colectiva en la creación de modelos para solucionar la selección de rasgos es cada vez más común, debido a las limitaciones de muchos métodos existentes producto de las características de complejidad de este problema. Existen, en la literatura, diversas soluciones a la selección de rasgos utilizando metaheurísticas poblacionales, específicamente optimización basada en colonias de hormigas, que incluyen el uso de Teoría de Conjuntos Aproximados.

2.1 Propuestas para la solución de la selección de rasgos

En 2003 Jensen y Chen propusieron un método de selección de rasgos (Jensen and Shen, 2003) que utiliza optimización con colonias de hormigas y conjuntos aproximados en la que el grafo se define como un conjunto de nodos, cada uno de los cuales representa un atributo de los datos. En este grafo, la feromona es almacenada en las aristas, y su valor está determinado por la dependencia entre los atributos en sus extremos, es decir, la cantidad de feromona en la arista (i, j) es una función del grado de pertenencia del atributo a_j respecto al atributo a_i. Este enfoque presenta el inconveniente de que la deseabilidad de seleccionar el atributo a_j en su camino por una hormiga depende del nodo a_i visitado anteriormente y no de los restantes atributos previamente seleccionados. Jensen en su tesis doctoral (Jensen, 2005) cambió esta perspectiva y asoció la feromona a los nodos.

Una de las propuestas más recientes y que sirvió de inspiración en la realización de este trabajo es el algoritmo ACO-RST-FSP de (Gómez Díaz, 2011). Este algoritmo es una variante de Sistema de colonia de hormigas, el cual usa la regla de transición proporcional pseudoaleatoria explicada anteriormente. Gómez Díaz asocia la feromona a los nodos del grafo y utiliza como medida heurística para un atributo a_i la calidad de la clasificación del conjunto de atributos visitados unido al atributo analizado $\eta_{i,j} = \gamma_{B^k \cup \{a_j\}}$. En este trabajo Gómez realiza un estudio de los parámetros del algoritmo, centrándose en β y q_0, ya que son estos los responsables de la fuerza que tiene la heurística en el algoritmo y de la relación exploración-explotación, lo que influye de manera significativa en la calidad de las soluciones encontradas y también en la

cantidad de estas. Del estudio de dichos parámetros Gómez recomienda utilizar $\beta = 1$ y $q_0 = 0.3$, pues fueron estos los que mejores resultados arrojaron.

2.2 Algoritmo ACO-RST-IT-FSP

A continuación se presenta un modelo que utiliza la variante *sistema de colonia de hormigas* de la optimización basada en colonias de hormigas como método de búsqueda de subconjuntos, la medida *ganancia de información* de la Teoría de la Información como función de evaluación heurística de la calidad de los subconjuntos y la medida *calidad de aproximación de la clasificación* de la Teoría de Conjuntos Aproximados como función de selección de reductos.

Figura 3 Modelo ACO-RST-IT-FSP

El algoritmo ACO-RST-IT-FSP se modela de la siguiente forma. Sea $S_d = (U, A \cup \{d\})$ un sistema de decisión sobre el que se desea correr el modelo, siendo $A = \{a_1, a_2, \dots, a_n\}$ el conjunto de los atributos condicionales, es decir, el conjunto que contiene todos los atributos que podrían formar parte de los reductos, d el atributo de decisión y $D = \{d\}$. Las características del modelo son las siguientes.

Grafo de feromonas

El grafo por el que se moverán las hormigas artificiales, llamado grafo de feromonas, ya que este solo contiene los valores de la feromona artificial depositada por las hormigas; es fuertemente conexo y está compuesto por n nodos, donde cada nodo representa un atributo del conjunto A. A diferencia del ACO tradicional, en la selección de rasgos se suele depositar la feromona en los nodos, denotada τ_i para el nodo i, y no en los arcos, siendo esta una medida de cuán deseado ha sido un atributo durante la ejecución del algoritmo, determinado por la dependencia de dicho atributo con respecto al resto de los atributos.

Construcción de las hormigas y asignación inicial

Al inicio de cada ciclo, las hormigas son creadas en una cantidad m dependiente del número de rasgos siguiendo un conjunto de reglas propuestas por (Bello et al., 2005), de forma que la función que determina el número de hormigas se define por:

$$f_h(n) = \begin{cases} n & n \in [1,19] \\ 24 & n \in [20,49] \wedge \frac{2}{3}n \leq 24 \\ \langle \frac{2}{3}n \rangle & n \in [20,49] \wedge \frac{2}{3}n > 24 \\ 33 & n \geq 50 \wedge \frac{n}{2} \leq 33 \\ \langle \frac{n}{2} \rangle & \text{en otro caso} \end{cases}$$

Donde $\langle x \rangle$ es el valor de x redondeado al entero más cercano. Esta función usa la clasificación de la escala del problema de selección de rasgos dada por (Kudo and Sklansky, 2000), la cual clasifica dichos problemas en tres clases: pequeña escala si $n \in [1,19]$, mediana escala si $n \in [20,49]$ y gran escala si $n \geq 50$.

Luego de creadas las hormigas, cada una es situada en un nodo siguiendo las reglas propuestas por (Bello et al., 2005) y presentadas a continuación:

1. Si $m < n$, se sitúa cada hormiga en un nodo aleatorio.
2. Si $m = n$, se le asigna a cada nodo una hormiga.

3. Si $m > n$, se sitúan n hormigas siguiendo la regla 2 y luego las restantes $m - n$ hormigas son situadas según la regla 1.

Procedimiento de transición

Estando ya las hormigas en sus posiciones iniciales se procede a la búsqueda de subconjuntos. Para esto, cada hormiga comienza a construir una solución, añadiendo un nuevo rasgo a su subconjunto B^k con cada nodo nuevo que visita. Inicialmente $B^k = \{n_1^k\}$, siendo $n_1^k = i$ el nodo inicial asignado a la hormiga k. Los nodos que puede visitar una hormiga son todos los que no hayan sido visitados $N^k = A - B^k$, es decir, todos los nodos que no estén presentes ya en su solución. La selección del próximo nodo a visitar se realiza de forma aleatoria donde cada nodo i tiene una probabilidad p_i^k de ser visitado por la hormiga k. Dichas probabilidades son asignadas mediante la siguiente función de transición:

1. Si $q \leq q_0$, esta es la regla proporcional pseudoaleatoria de ACS:

$$p_i^k = \begin{cases} 1 & i \in N^k \wedge \forall_{j \in N^k} \tau_i \left(IG_{B^k \cup i}(D)\right)^\beta \geq \tau_j \left(IG_{B^k \cup j}(D)\right)^\beta \\ 0 & \text{en otro caso} \end{cases}$$

2. Si $q > q_0$:

$$p_i^k = \begin{cases} \dfrac{\tau_i \left(IG_{B^k \cup i}(D)\right)^\beta}{\sum_{j \in N^k} \tau_j \left(IG_{B^k \cup j}(D)\right)^\beta} & i \in N^k \\ 0 & i \notin N^k \end{cases}$$

   Donde $IG_{B^k \cup i}(D)$ es el valor de la medida ganancia de información para el posible nuevo subconjunto solución de la hormiga, o sea, este es el valor de la medida heurística que tendría el subconjunto B^k si la hormiga diera el paso hacia el nodo i y lo incorporara a su solución.

Criterio de parada de las hormigas

Una hormiga detiene su movimiento cuando ha encontrado un conjunto de atributos tal que la medida calidad de la clasificación sea la máxima alcanzable para ese conjunto de datos, por ende, una hormiga continuará su movimiento mientras no se cumpla que $\gamma_{B^k}(D) = \gamma_A(D)$;

nótese que esto es equivalente a decir que una hormiga se mueve mientras no encuentre un superreducto o le queden nodos por visitar.

Búsqueda local

Luego de que una hormiga encuentra un $B \subset A$ que cumple con $\gamma_B(D) = \gamma_A(D)$ se realiza un intento de reducir nodo a nodo la longitud del mismo, para garantizar que lo que se obtenga sea un reducto, ya que la construcción por adición de las hormigas garantiza la obtención de superreductos (Yao et al., 2006). Si $\exists_{R \subset B}\ \gamma_R(D) = \gamma_B(D) = \gamma_A(D)$ entonces B no es un reducto por definición, por tanto, en vez de considerar B se considerará R. Luego a R se le hará lo mismo hasta que sea imposible eliminar algún atributo manteniendo la calidad del subconjunto. De esta forma se puede garantizar que el subconjunto final es un reducto.

Actualización fuera de línea de la feromona

Al final de cada ciclo, como indica la variante ACS, se realiza una actualización de la feromona en los nodos pertenecientes al mejor reducto encontrado hasta el momento como indica la fórmula:

$$\forall_{i \in S_{\text{mejor-global}}}\ \tau_i \leftarrow (1-\rho)\tau_i + \rho \frac{1}{|S_{\text{mejor-global}}|}$$

La cantidad adicionada es el recíproco de la cardinalidad del mejor reducto encontrado para que mientras más corto sea este, mayor la cantidad de feromona adicionada.

Criterio de parada de la colonia

El criterio de parada de un algoritmo ACO depende del problema y del objetivo que se quiere lograr, en general existen diversos criterios (Dorigo and Stutzle, 2003). Dado que el algoritmo aquí descrito tiene como meta encontrar la mayor cantidad posible de reductos y tan cortos como se pueda se eligió como criterio de parada cuando la ejecución complete un número determinado de iteraciones o ciclos.

Algoritmo ACO-RST-IT-FSP

Entrada: Conjunto de datos en forma de sistema de decisión.

Salida: Colección de reductos.

Paso 1: Inicializar conjunto de reductos $S = \emptyset$

Paso 2: Inicializar rastros de feromona τ_i

Paso 3: para $ciclo \leftarrow 1 \dots ciclo_{\max}$

Paso 3.1: Crear hormigas, inicializarlas y distribuirlas

Paso 3.1.1: para $k \leftarrow 1 \dots m$

Paso 3.1.1.1: Seleccionar nodo inicial i para hormiga k

Paso 3.1.1.2: $B^k \leftarrow \{i\}$

Paso 3.2: para $k \leftarrow 1 \dots m$

Paso 3.2.1: mientras $\gamma_{B^k}(D) \neq \gamma_A(D)$

Paso 3.2.1.1: seleccionar nuevo nodo i a visitar por la hormiga k según la función de transición

Paso 3.2.1.2: $B^k \leftarrow B^k \cup \{i\}$

Paso 3.2.1.3: $\tau_i \leftarrow (1-\varphi)\tau_i + \varphi\tau_0$

Paso 3.2.2: Si $B^k \subset A$ entonces

Paso 3.2.2.1: R recibe el resultado de intentar reducir B mediante una búsqueda local.

Paso 3.2.2.2: $S \leftarrow S \cup \{R\}$

Paso 3.3: Seleccionar el reducto R más corto en S

Paso 3.4: para cada nodo $i \in R$

Paso 3.4.1: $\tau_i \leftarrow (1-\rho)\tau_i + \rho\frac{1}{|R|}$

Fin del algoritmo

2.3 Análisis de la complejidad computacional de ACO-RST-IT-FSP

Recientemente el análisis del tiempo de ejecución de la metaheurística ACO ha sido estudiado, pero los esfuerzos han sido limitados a clases específicas de problemas o a versiones simplificadas del algoritmo, en particular al estudio de una pieza del algoritmo como la influencia de la feromona (Doerr et al., 2007, Neumann and Witt, 2006). Sin embargo, en ACO aplicado a la selección de rasgos el tiempo de ejecución está determinado fundamentalmente por el costo computacional de la evaluación heurística.

Siendo c la cantidad de iteraciones o ciclos de ejecución del algoritmo, m la cantidad de hormigas y n el número de rasgos presentes en el sistema de decisión que contiene los datos. La cantidad de evaluaciones que tendrá que hacer el algoritmo ACO-RST-IT-FSP asumiendo el peor caso, este es, cuando no se encuentra ningún reducto y las hormigas hacen el recorrido de los n nodos:

$$e(c,m,n) = cm\sum_{i=1}^{n-1}(n-i)$$

Esto se explica de la siguiente manera. El algoritmo ejecutará c ciclos, en cada ciclo m hormigas harán su recorrido, y cada hormiga en su recorrido dará, en el peor caso, $n-1$ pasos, asumiendo que el posicionamiento inicial no cuenta como uno. En el primer paso cada hormiga tiene $n-1$ nodos que verificar, por lo que tiene que hacer igual cantidad de evaluaciones, en el segundo paso ya tiene un nodo menos que analizar, por lo que tiene que hacer $n-2$ evaluaciones, así sucesivamente hasta el último paso, en el que tiene que hacer sólo una evaluación.

El cálculo de la función heurística ganancia de información está limitado en tiempo por el costo de hallar la partición que produce un subconjunto de atributos, y el tiempo de hallar esta es:

$$f(u,r) = ku^2r$$

Donde k es una constante positiva, u es el total de objetos en el sistema y r es la cantidad de rasgos involucrados en la evaluación. Esto se debe a que para construir la partición es necesario

comparar todos los objetos por pares, y para diferenciar un par de objetos se necesitan comparar los r rasgos.

Volviendo a la cantidad de evaluaciones se puede decir que el tiempo de ejecución del algoritmo está principalmente determinado por el número de comparaciones. Por lo tanto, el tiempo dedicado en el algoritmo a hacer comparaciones atributo-objeto es de:

$$t(c,m,n,u) = cm\sum_{i=1}^{n-1}(n-i)f(u,i+1)$$

En el primer paso cada hormiga evalúa la función heurística para dos rasgos, en el segundo paso, lo hace para tres rasgos, y así sucesivamente, de ahí el factor $f(u,i+1)$. Sustituyendo f se obtiene:

$$t(c,m,n,u) = kcmu^2\sum_{i=1}^{n-1}(n-i)(i+1)$$

$$= \frac{1}{6}kcmu^2(n^3+3n^2-4n)$$

De este resultado se puede concluir que el tiempo de corrida del algoritmo ACO-RST-IT-FSP presenta un crecimiento cúbico con respecto a la cantidad de rasgos, cuadrático respecto a la cantidad de objetos y lineal respecto a la cantidad de iteraciones y hormigas, y como las variables presentes en el análisis son independientes, el tiempo de ejecución del algoritmo es un $O(cmu^2n^3)$. Este tiempo se reduce sustancialmente si el sistema presenta reductos, ya que las hormigas no tendrán que realizar el recorrido completo. Algo que queda por analizar en este problema es la influencia que tiene la feromona en este tiempo, lo cual es un problema de gran dificultad debido a la naturaleza estocástica de ACO.

Estrategia de reducción del tiempo de ejecución

Una técnica empleada para reducir el tiempo de ejecución del algoritmo fue la utilización de una estructura de datos que ayuda a reducir la cantidad de veces que se calcula la función heurística. Específicamente se utilizó una tabla hash que almacena los pares subconjunto-calidad siempre que este se calcula; así, cada vez que se necesita el valor de calidad para un

subconjunto se busca primero este par en la tabla, si está presente, este se devuelve, si no, entonces se calcula y se almacena. Esta estrategia es práctica para conjuntos de datos que no sean extremadamente grandes en cuanto a cantidad de atributos, téngase en cuenta que la cantidad máxima de subconjuntos que se evaluarán está acotada por $e(c,m,n) \leq \frac{1}{2}cm(n^2-n)$. Nótese también que el tiempo de búsqueda y acceso a un elemento en una tabla hash es un $O(1)$, enormemente menor que calcular la función de calidad.

Utilizando esta misma idea Gómez evaluó el efecto de la estrategia mostrando su eficacia. Para el conjunto de datos *Dermatology* del repositorio UCI se obtuvo la siguiente tabla.

Tabla 1 Número de subconjuntos evaluados por ciclo

Ciclo (i)	Calculados (ii)	Recuperados de la estructura (iii)	Total de subconjuntos requeridos (iv)
1	11467	842	12309
2	9918	2994	12912
3	8736	4208	12944
4	8943	3940	12883
5	8063	4950	13013
6	8553	4484	13037
7	8102	4815	12917
8	6754	6158	12912
9	8648	4377	13025
10	6709	6217	12926
11	8326	4723	13049
12	6274	6700	12974
13	6469	6581	13050
14	5993	6836	12829
15	7001	6009	13010

El siguiente gráfico representa la información de la Tabla 1 Número de subconjuntos evaluados por ciclo. La columna ii de la tabla, graficada como una línea de puntos, representa el número de veces que la función heurística fue evaluada por la expresión. La línea de trazos, columna iii de la tabla, representa el número de veces que no fue necesario recalcular la función ya que su valor se encontraba almacenado en la estructura de datos. La línea continua, columna iv de la tabla, representa la totalidad de veces que se requirió el valor de la función heurística. El

número total de candidatos que necesitan ser evaluados por ciclo muestra un comportamiento oscilatorio debido al carácter estocástico de la metaheurística ACO.

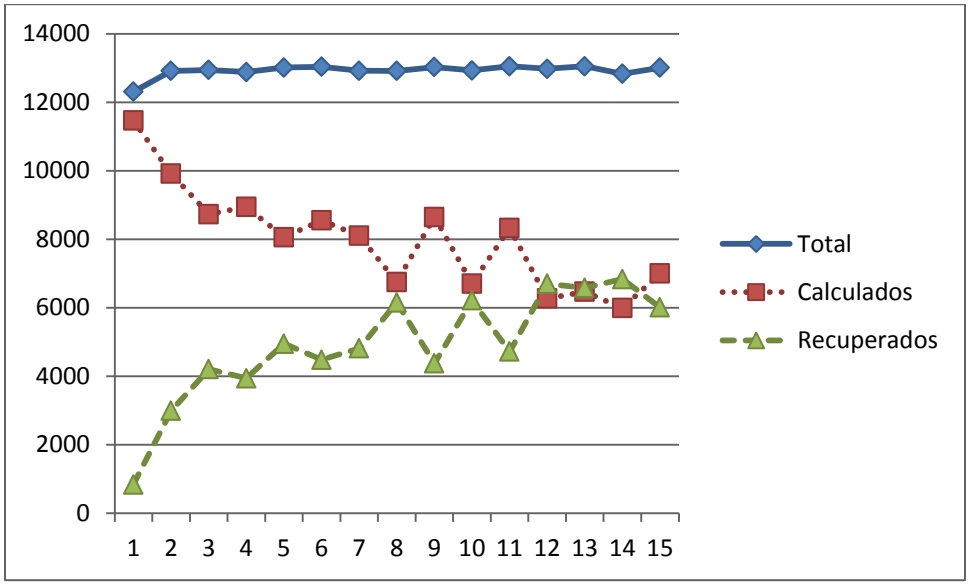

Figura 4 Evaluaciones de subconjuntos

2.4 Solución al problema distribuido de selección de rasgos

El problema de la selección de rasgos sobre múltiples fuentes de datos, conocido también como problema de selección de rasgos en contexto distribuido, es, en términos generales, muy similar al problema clásico de selección de rasgos. Esta nueva problemática surge cuando los datos se encuentran separados física o virtualmente y se necesita realizar la selección de rasgos de manera que todas las fuentes de datos intervengan en el análisis, pero de manera que no se conozcan fuera de su ámbito local.

Gómez en su tesis de doctorado (Gómez Díaz, 2011) propone un modelo, con excelentes resultados, para darle solución al problema de selección de rasgos para múltiples fuentes de datos. El algoritmo por él propuesto, llamado D.ACO-RST-FSP establece un método de comunicación entre los subsistemas mediante el intercambio de metadatos[5], de esta forma se

[5] Estos son datos acerca de los datos.

conserva la privacidad de los datos en cada fuente. Este modelo consiste en poner a correr en cada subsistema una colonia de hormigas, y estas, cada una cierta cantidad de ciclos, realizaran un intercambio de sus grafos de feromonas. Por consiguiente, cada colonia tiene dos grafos, uno representando la feromona depositada por las hormigas locales τ_i y otro que simboliza el promedio de la feromona de las restantes colonias ϕ_i, este último es un reflejo del conocimiento adquirido por las colonias vecinas de sus datos locales. Este nuevo término es incorporado a la función de transición utilizada en el algoritmo local en forma de un nuevo factor con su exponente γ para indicar la sensibilidad al *conocimiento extranjero*:

$$p_i^k = \begin{cases} \dfrac{\tau_i \left(\gamma^v_{B^k \cup i}(D)\right)^\beta (\phi_i)^\gamma}{\sum_{j \in N^k} \tau_j \left(\gamma^v_{B^k \cup j}(D)\right)^\beta (\phi_i)^\gamma} & i \in N^k \\ 0 & i \notin N^k \end{cases}$$

2.5 Algoritmo D.ACO-RST-IT-FSP

El modelo presentado para resolver dicho problema trata con varios sistemas de datos homogéneos, es decir, que son compatibles los sistemas de decisión de las diversas fuentes de datos, cada sistema posee los mismos atributos y el mismo esquema, sólo se diferencian en los datos que contienen.

Para darle solución a la selección de rasgos en contexto distribuido es necesario crear un modelo formado por varios subsistemas, capaz de cooperar entre ellos intercambiando metainformación sin comprometer la privacidad de los datos (Ver Figura 5 Esquema de funcionamiento de D.ACO-RST-IT-FSP). Siguiendo estos preceptos, propone el algoritmo D.ACO-RST-IT-FSP inspirado en el algoritmo D.ACO-RST-FSP de (Gómez Díaz, 2011) y en la variante multitipo del ACS de (Nowé et al., 2004).

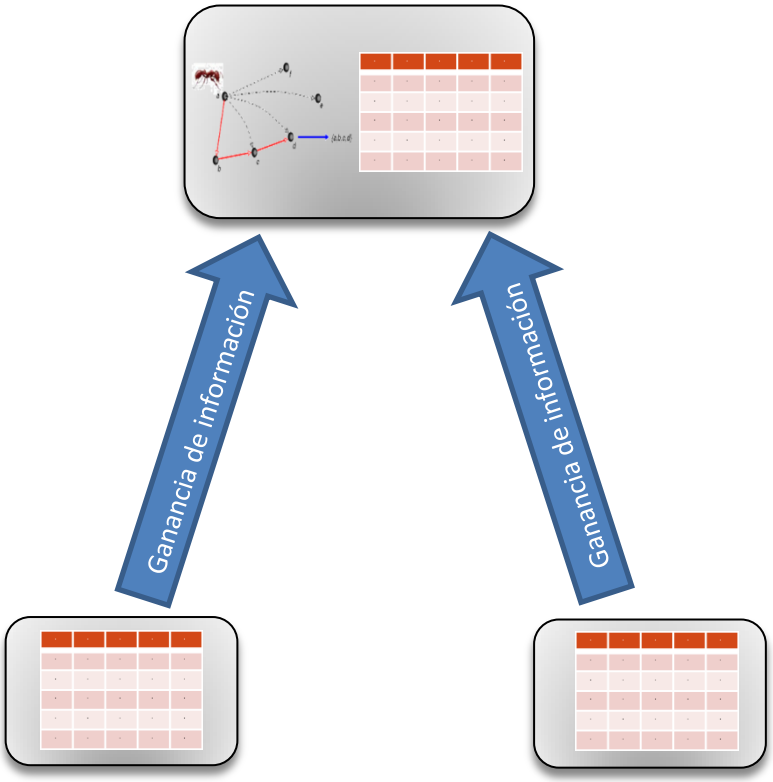

Figura 5 Esquema de funcionamiento de D.ACO-RST-IT-FSP

Esta variante distribuida presenta un comportamiento similar a su original para contexto local. Sean dados $S_1, S_2, \dots, S_u$ sistemas de decisión homogéneos e independientes y se quiere hallar reductos en el subsistema S_v $1 \leq v \leq u$, se procede de la siguiente manera. Se coloca un algoritmo D.ACO-RST-IT-FSP a correr en el subsistema S_v, llamado subsistema principal, y un

algoritmo en cada uno de los restantes subsistemas capaz de aceptar peticiones de cálculos de ganancia de información y enviarlos al subsistema principal.

Los cambios necesarios para aceptar y asimilar los valores heurísticos enviados por los subsistemas *vecinos* se reflejan en la función de transición como sigue:

1. Si $q \leq q_0$:

$$p_i^k = \begin{cases} 1 & i \in N^k \wedge \forall_{j \in N^k} \tau_i \left(IG^v_{B^k \cup i}(D)\right)^\beta \left(\chi_{B^k \cup i}\right)^\gamma \geq \tau_j \left(IG_{B^k \cup j}(D)\right)^\beta \left(\chi_{B^k \cup j}\right)^\gamma \\ 0 & \text{en otro caso} \end{cases}$$

2. Si $q > q_0$:

$$p_i^k = \begin{cases} \dfrac{\tau_i \left(IG^v_{B^k \cup i}(D)\right)^\beta \left(\chi_{B^k \cup i}\right)^\gamma}{\sum_{j \in N^k} \tau_j \left(IG^v_{B^k \cup j}(D)\right)^\beta \left(\chi_{B^k \cup j}\right)^\gamma} & i \in N^k \\ 0 & i \notin N^k \end{cases}$$

Donde $\chi_{B^k \cup i}$ es el valor promedio de las ganancias de información pertenecientes a los subsistemas vecinos.

$$\chi_{B^k \cup i} = \frac{\sum_{w=1}^{v-1} IG^w_{B^k \cup i}(D) + \sum_{w=v+1}^{u} IG^w_{B^k \cup i}(D)}{u - 1}$$

El parámetro γ representa el grado de aceptación que tendrán las hormigas de las ganancias de información de los subsistemas ajenos. Para $\gamma = 0$ la colonia simplemente ignorará esta metainformación y funcionará como el algoritmo puramente local. Para mayores valores de γ, el algoritmo incrementa su dependencia de la metainformación enviada por los subsistemas vecinos.

2.6 Consideraciones parciales

De lo visto hasta ahora puede concluirse que la aplicación de la metaheurística ACO a un modelo de selección de rasgos para múltiples fuentes de datos es una técnica válida, combinada con la estrategia de cooperación adecuada y un diseño eficaz del grafo de feromonas.

El uso de la medida ganancia de información de la Teoría de la Información como metadato de intercambio entre los subsistemas permite la construcción de un algoritmo de selección de rasgos en contexto distribuido.

3 Implementación y evaluación de los algoritmos

La implementación de los algoritmos ACO-RST-IT-FSP y D.ACO-RST-IT-FSP fue llevada a cabo en el lenguaje de programación C# para *.NET Framework 4.0*. La cooperación de los subsistemas distribuidos se logró mediante el uso de WCF (*Windows Communication Foundation*), un modelo de programación para la creación de aplicaciones orientadas a servicios (Lowy, 2007). La siguiente figura muestra la arquitectura básica del funcionamiento de la comunicación entre los diversos subsistemas que intervienen en la selección de rasgos.

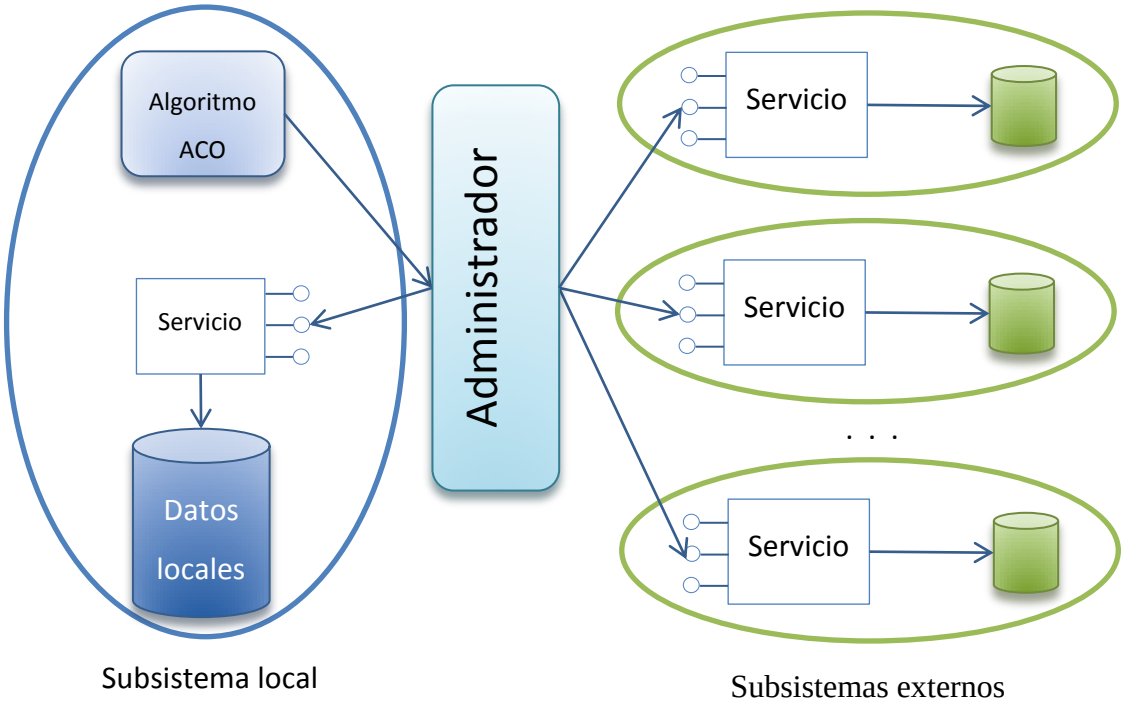

Figura 6 Arquitectura de la comunicación entre los subsistemas. Modelo D.ACO-RST-IT-FSP

Como puede verse, cada servicio es el único capaz de acceder a los datos de su subsistema y tiene la responsabilidad de aceptar y responder peticiones de valores heurísticos. Ver el siguiente diagrama de clases simplificado para información más detallada de las dependencias entre los elementos del modelo.

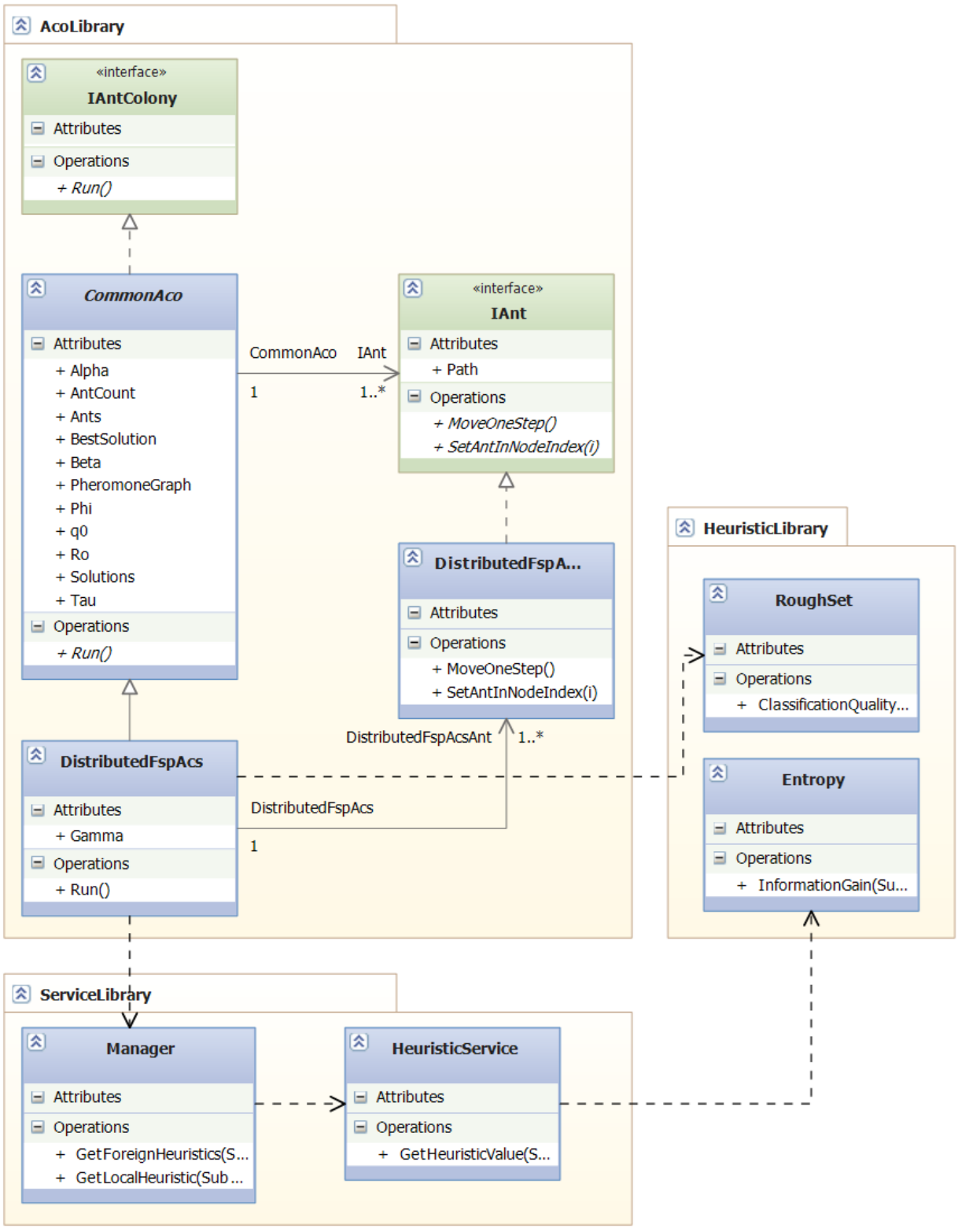

Figura 7 Diagrama de clases simplificado del modelo D.ACO-RST-IT-FSP

A continuación se realizará una evaluación de los algoritmos desarrollados a lo largo de la investigación. Esta evaluación se llevará a cabo mediante la comparación con otros modelos diseñados para resolver el mismo problema, utilizando seis conjuntos de datos del repositorio UCI, el cual contiene una gran cantidad de datos de problemas reales.

Para poder aplicar los algoritmos, los atributos numéricos presentes en los datos fueron discretizados, esto se debe a que los conceptos utilizados de la Teoría de la Información y la Teoría de Conjuntos Aproximados están concebidos para atributos discretos. Consultar el Anexo 1 Características de los conjuntos de datos para más información sobre las bases de casos utilizadas.

3.1 Validación del algoritmo ACO-RST-IT-FSP

Esta validación se hizo comparando el algoritmo ACO-RST-IT-FSP con modelos existentes en la bibliografía. Los resultados de los algoritmos con los que se compara son tomados de la tesis de doctorado (Gómez Díaz, 2011). El criterio de comparación elegido fue la cardinalidad del reducto más corto. Las bases de casos elegidas para realizar la comparación entre algoritmos en contexto local fueron Breast-Cancer-Wisconsin, HeartJen, LedJen, Dermatology y Lung-Cancer. La razón de esta elección se debe a la cantidad de atributos que presentan; las dos primeras contienen 8 y 13 atributos predictivos respectivamente, lo que las sitúa en la escala pequeña de un problema de selección de rasgos, la tercera contiene 24 atributos y la cuarta tiene 34 para elegir, lo que las sitúa en la escala mediana, y por último Lung-Cancer posee 56 atributos, sobrepasando el límite de 49, por lo que pertenece a la escala grande. Con esta selección puede analizarse el efecto que tiene la cantidad de rasgos en la calidad de las soluciones obtenidas por cada método, pues la cantidad de rasgos presentes en los datos determina el tamaño del espacio de búsqueda sobre el que deben trabajar los algoritmos. En el presente análisis se mostrarán los resultados para tres de los conjuntos seleccionados, uno de cada clasificación. Los resultados de los restantes conjuntos de datos pueden encontrarse en el Anexo 3 Comparaciones entre varios algoritmos en contexto local.

Debe quedar claro que una de las ventajas de los algoritmos poblacionales es que pueden encontrar más de un reducto, e inclusive, más de una vez un reducto de cardinalidad mínima.

Por esta razón fue omitida esta parte del resultado en el presente estudio, ya que se hacen comparaciones con algunos modelos que sólo son capaces de encontrar un reducto.

SAVGeneticReducer es un algoritmo que encuentra reductos utilizando Algoritmos Genéticos como método de búsqueda, ha sido evaluado tomando la implementación del sistema ROSETTA (Øhrn et al., 1998, Øhrn, 1999).

Las características de los métodos *Best-First*, *Greedy Stepwise* y *Genetic Algorithms* son descritos en (Witten and Frank, 2005), y su implementación ha sido tomada de WEKA (Hall et al., 2009); estos métodos de búsqueda se combinaron con dos funciones de evaluación de subconjuntos (1-CfsSubsetEval, 2-ConsistencySubsetEval) obteniéndose 6 variantes.

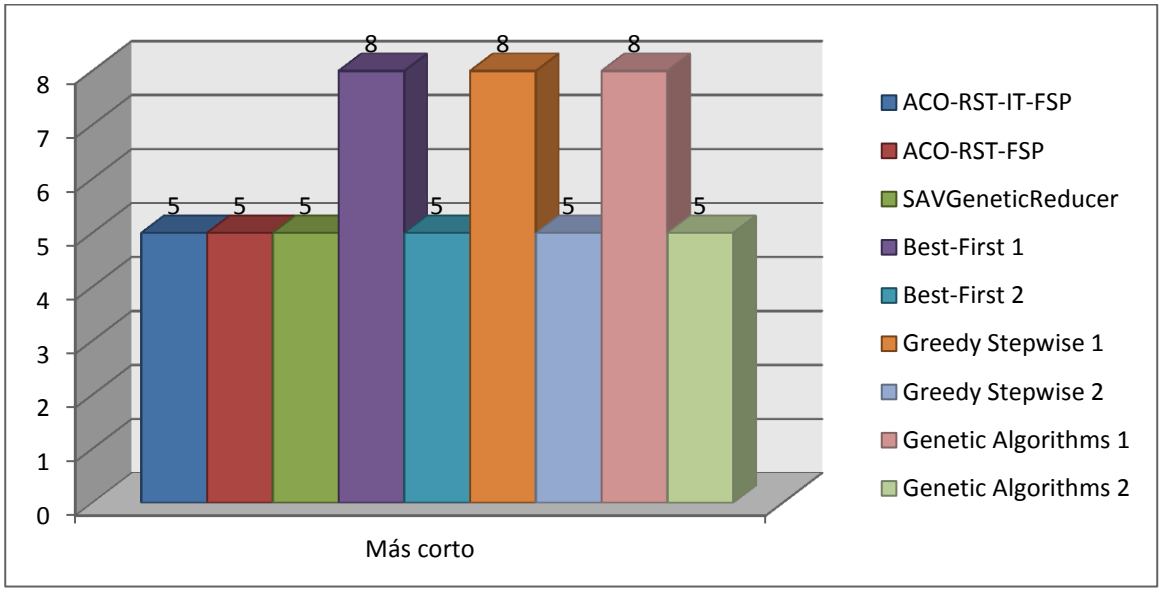

Figura 8 Comparación entre los algoritmos para Breast-Cancer-Wisconsin

En la Figura 8 se puede apreciar que la cardinalidad del mejor reducto encontrado para el conjunto de datos Breast-Cancer-Winsconsin es 5 y que fue encontrado por varios métodos, incluyendo ACO-RST-IT-FSP, el cual muestra resultados que igualan a los mejores obtenidos.

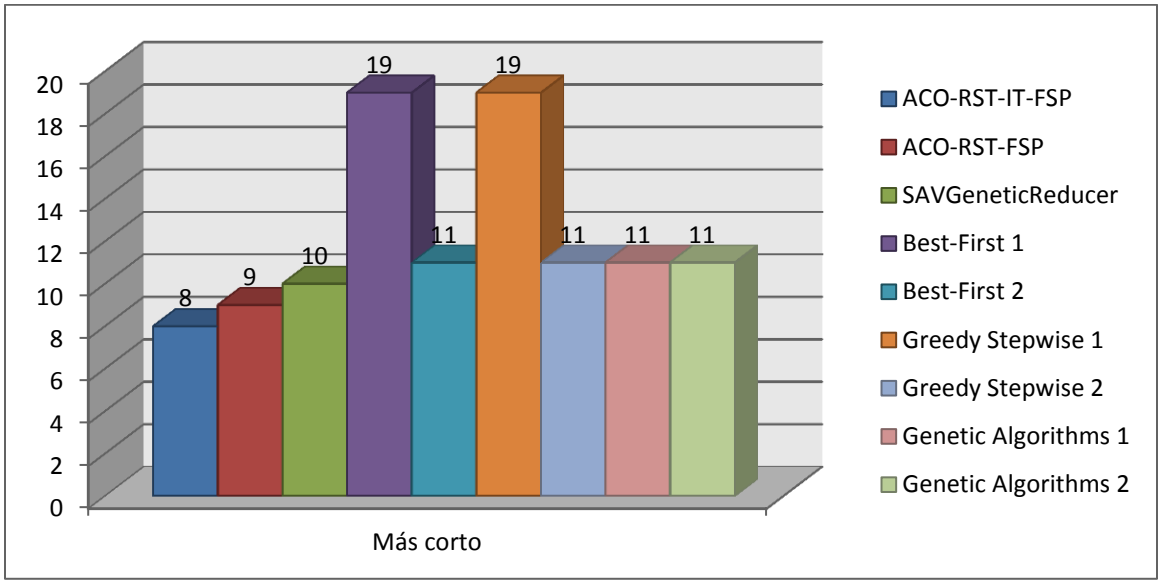

Figura 9 Comparación entre los algoritmos para Dermatology

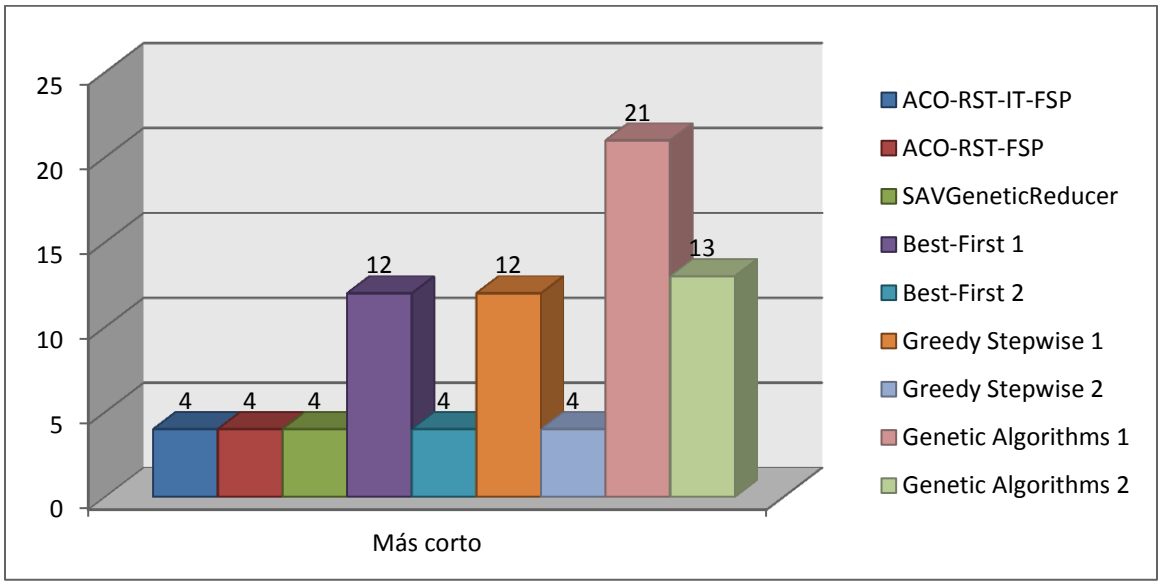

Figura 10 Comparación entre los algoritmos para Lung-Cancer

En la Figura 9, que muestra los resultados obtenidos por los mismos algoritmos para el conjunto de datos Dermatology puede apreciarse que el mejor reducto, este de longitud 8, fue encontrado por el método ACO-RST-IT-FSP, y además fue el único método que encontró un reducto de longitud 8.

La Figura 10 revela los resultados que se obtuvieron utilizando el conjunto de datos Lung-Cancer. En esta puede notarse que el reducto más corto encontrado tiene cardinalidad igual a cuatro y fue encontrado por las dos variantes de ACO presentadas, el algoritmo SavGeneticReducer, Best-First 2 y Greedy-Stepwise 2.

Algo notable en los resultados analizados es que siempre los algoritmos ACO mostraron un comportamiento mejor o igual que los restantes modelos, además de que se puede observar que sus resultados son superiores a medida que aumenta la cantidad de rasgos, en relación con otros algoritmos y atendiendo al criterio elegido; lo que reafirma la idea de que estos se cuentan entre los modelos más eficaces para resolver el problema de la selección de rasgos.

3.2 Validación del algoritmo D.ACO-RST-IT-FSP

Para evaluar el algoritmo desarrollado en la investigación en su variante para múltiples fuentes de datos se procedió a elegir varios conjuntos de datos del repositorio UCI, para luego dividirlos en particiones de forma que se pueda simular el ambiente distribuido, construyendo un escenario de prueba desde el punto de vista teórico-práctico siguiendo la idea expuesta en (Jasso-Luna et al., 2008, Martens et al., 2006). Cada conjunto de datos fue dividido en tres partes de manera que cada parte simula un subconjunto de datos separado de los restantes subconjuntos. Los datos fueron divididos usando el filtro *Stratified Remove Folds* de Weka, el cual mantiene la distribución de los objetos por clase en cada subconjunto de datos.

La evaluación del algoritmo D.ACO-RST-IT-FSP se hizo comparando los resultados obtenidos por este, contra los resultados obtenidos al aplicar la variante local a cada uno de los subconjuntos involucrados. Esto se hace para comparar la efectividad del algoritmo distribuido con la lograda

al hacer la selección de rasgos de no existir un algoritmo que funcione para múltiples fuentes de datos.

La otra comparación que se realizó para este algoritmo fue con la variante distribuida D.ACO-RST-FSP propuesto por (Gómez Díaz, 2011), por ser uno de los más recientes algoritmos que resuelven este problema en contexto distribuido, que además usa optimización basada en colonias y Teoría de Conjuntos Aproximados. Otra razón por la que este fue elegido es la buena calidad en las soluciones que obtuvo, pues demostró ser competitivo en la solución del problema.

Los resultados fueron obtenidos y combinados utilizando la media aritmética de los indicadores de salida de un total de 10 corridas para los conjuntos pequeños y 5 corridas para los más grandes. Los valores elegidos para los parámetros en las ejecuciones de ambos algoritmos fueron los sugeridos por Gómez Díaz en su tesis, $\beta = 5$, $q_0 = 0.9$ y $\gamma = 1.1$. Tanto el parámetro β como q_0 tienen valores obtenidos de un estudio realizado por (Gómez Díaz, 2011) con el objetivo de encontrar la configuración que mejores resultados arroja. Sin embargo, el parámetro γ, encargado de definir la sensibilidad de la colonia al conocimiento externo, no fue resultado de un análisis y su valor fue elegido arbitrariamente.

Los criterios elegidos para realizar la comparación fueron: 1) la longitud del reducto más corto, ya que este es el parámetro más apreciado generalmente en la selección de rasgos, 2) cantidad de veces que se encuentran reductos de cardinalidad mínima y 3) cantidad total de reductos encontrados. Dada la similitud entre las tres variantes presentes en la comparación el primer criterio siempre terminó en empate, lo que hace que se comparen en cuanto a los criterios 2) y 3).

Breast-Cancer-Wisconsin

A continuación se muestran dos gráficos de barras con estructuras similares. Ambos tienen tres grupos de tres barras cada uno, donde cada grupo representa un subsistema, es decir, los resultados obtenidos en el subconjunto de datos correspondiente. La primera barra de cada grupo representa el indicador obtenido por el algoritmo D.ACO-RST-IT-FSP, esta es la variante distribuida del algoritmo desarrollado en este trabajo. La segunda barra representa la variante

local del mismo algoritmo, ejecutado en contexto local sin conocimiento de la existencia de los otros subconjuntos y la tercera barra representa el indicador obtenido por el algoritmo D.ACO-RST-FSP elegido como punto de comparación para el contexto distribuido.

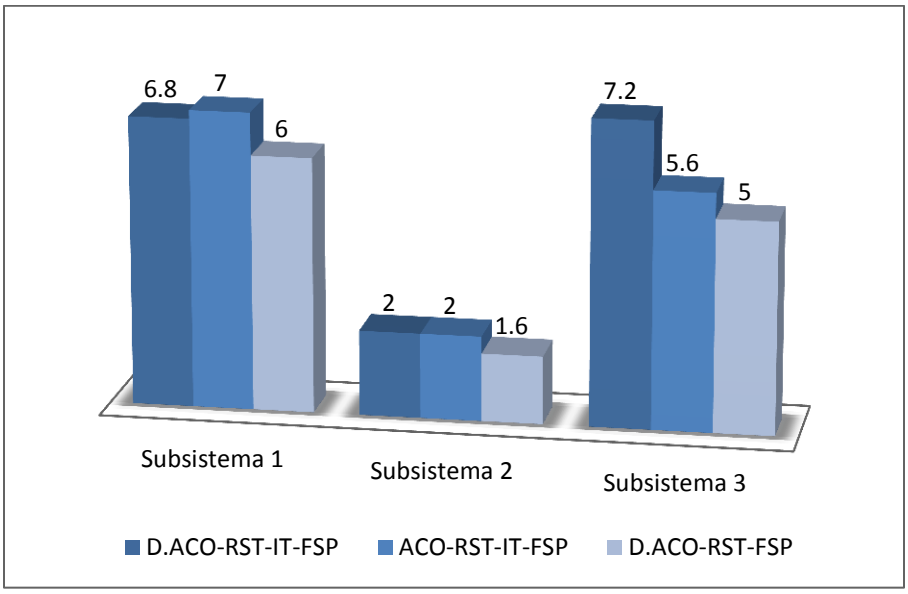

Figura 11 Comparación de los tres modelos para el conjunto de datos Breast-Cancer-Wisconsin. (Criterio 2)

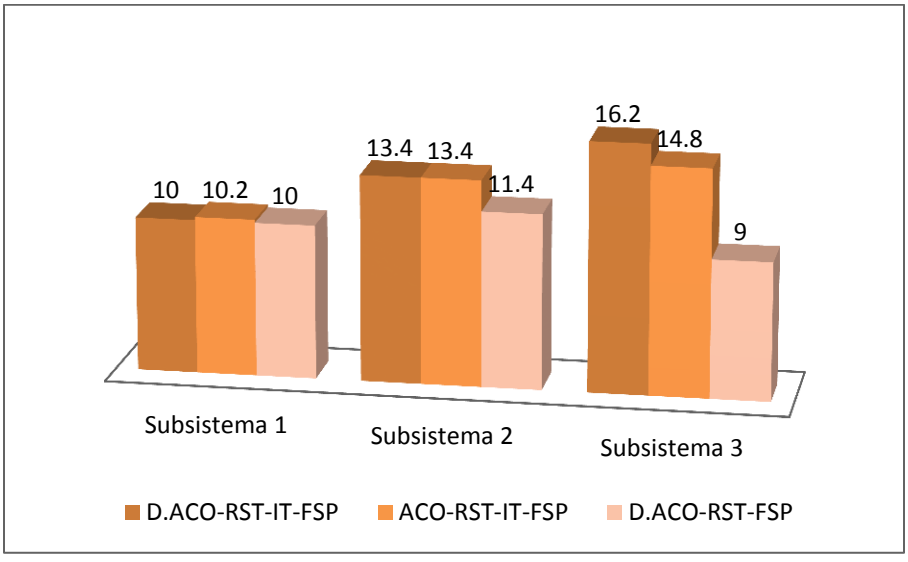

Figura 12 Comparación de los tres modelos para el conjunto de datos Breast-Cancer-Wisconsin. (Criterio 3)

La Figura 11 Comparación de los tres modelos para el conjunto de datos Breast-Cancer-Wisconsin. (Criterio 2) muestra los valores de la cantidad de reductos de longitud mínima

encontrados por cada uno de los algoritmos en cada uno de los subsistemas. Como se puede apreciar, la variante distribuida del nuevo algoritmo estudiado logra un desempeño menor que su hermano local en el primer subconjunto, con una diferencia de 0.2. En el segundo subconjunto se obtiene un empate, sin embargo, en el tercer subconjunto el algoritmo distribuido logra aventajar al local por 1.6. En cuanto a la comparación entre los algoritmos distribuidos, el presentado en esta investigación obtiene resultados superiores al algoritmo en el cual está inspirado, ACO-RST-FSP, en todos los subconjuntos de datos.

La Figura 12 Comparación de los tres modelos para el conjunto de datos Breast-Cancer-Wisconsin. (Criterio 3) muestra una comparación en cuanto a la cantidad total de reductos encontrados. Del análisis de este gráfico se puede notar un relativo empate entre los tres algoritmos en el primer subsistema, sobresaliendo muy ligeramente el algoritmo local sobre ambas variantes distribuidas. En el segundo subsistema D.ACO-RST-IT-FSP y su homólogo local logran un empate sobresaliendo ambos sobre el algoritmo distribuido de Gómez Díaz en 2 puntos, eso significa que los promedios del total de reductos del algoritmo ACO-RST-IT-FSP sobrepasan en 2 a los encontrados por ACO-RST-FSP.

Dermatology

Para esta partición del conjunto de datos Dermatology se obtuvieron los siguientes resultados.

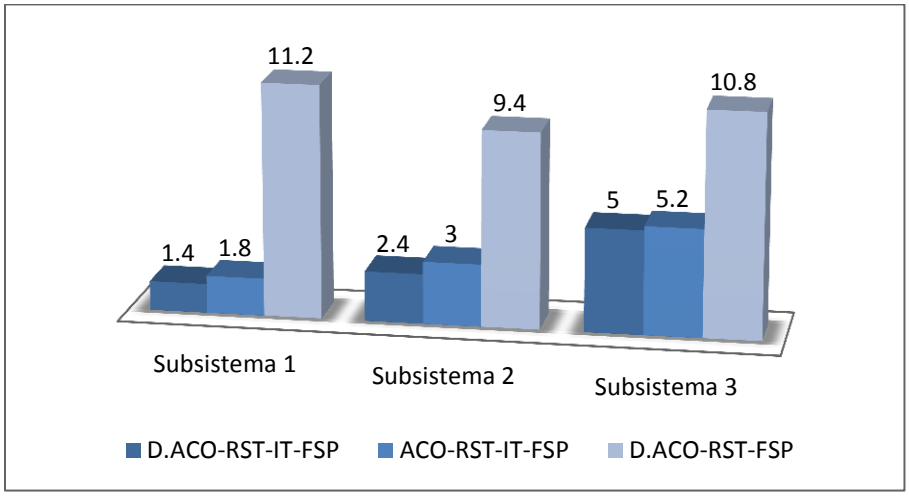

Figura 13 Comparación de los tres modelos para el conjunto de datos Dermatology. (Criterio 2)

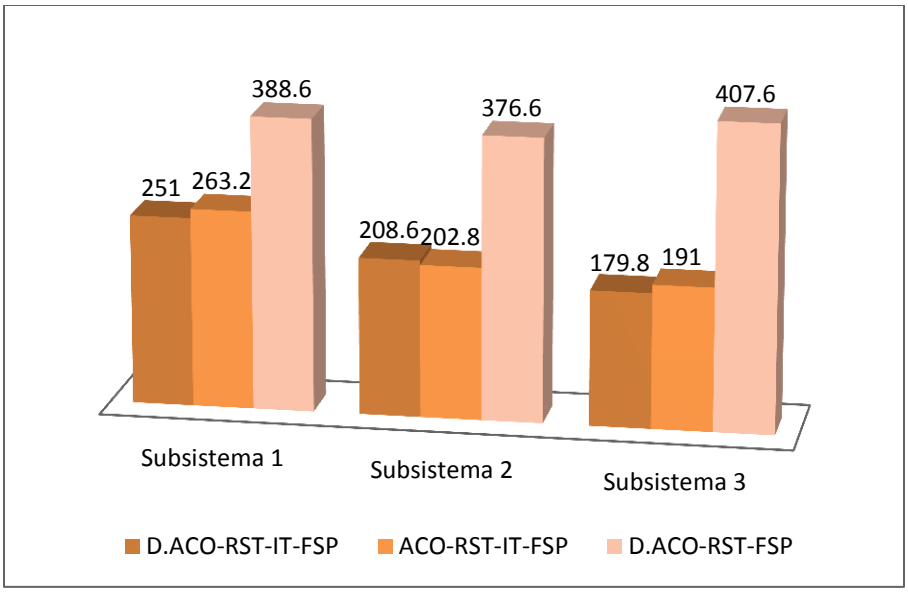

Figura 14 Comparación de los tres modelos para el conjunto de datos Dermatology. (Criterio 3)

Los algoritmos para la partición utilizada del conjunto de datos Dermatology mostraron un comportamiento completamente diferente obteniendo resultados muy similares las variantes local y distribuida de ACO-RST-IT-FSP, siendo ambas aventajadas drásticamente por el algoritmo D.ACO-RST-FSP.

Ledjen

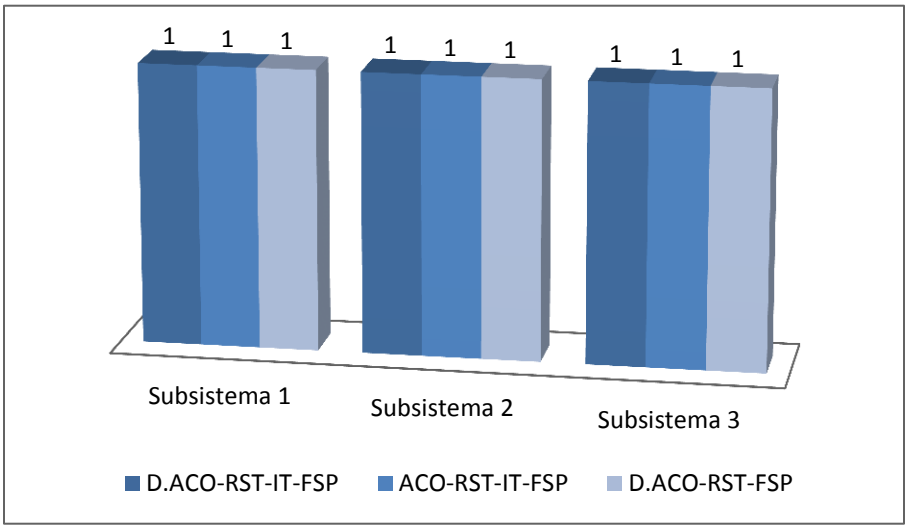

Figura 15 Comparación de los tres modelos para el conjunto de datos Ledjen. (Criterio 2)

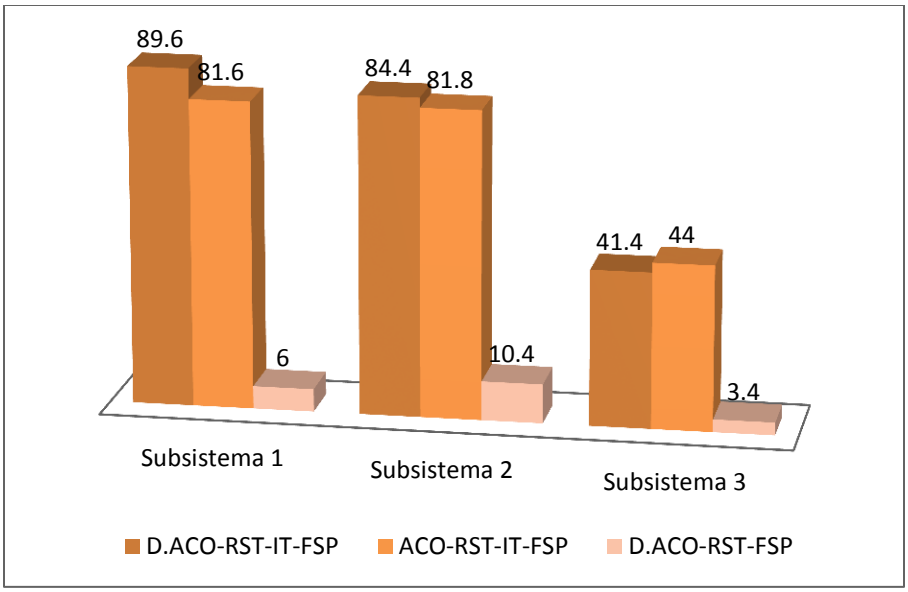

Figura 16 Comparación de los tres modelos para el conjunto de datos Ledjen. (Criterio 3)

Para este conjunto los resultados arrojados son contrarios a los obtenidos para Dermatology. Aunque según la cantidad de reductos minimales encontrados es exactamente uno para todos los algoritmos, debido a la presencia de un único reducto de longitud 5 muy fácil de encontrar en estos datos; cuando se compara en cuanto a la cantidad de reductos encontrados, ambos modelos desarrollados en esta tesis logran un desempeño muy superior al logrado por el algoritmo ACO-RST-FSP.

Vehicle

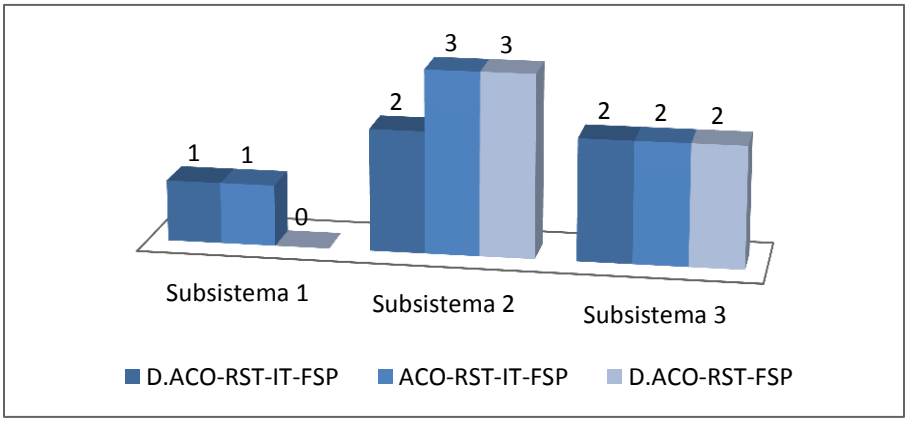

Figura 17 Comparación de los tres modelos para el conjunto de datos Vehicle. (Criterio 2)

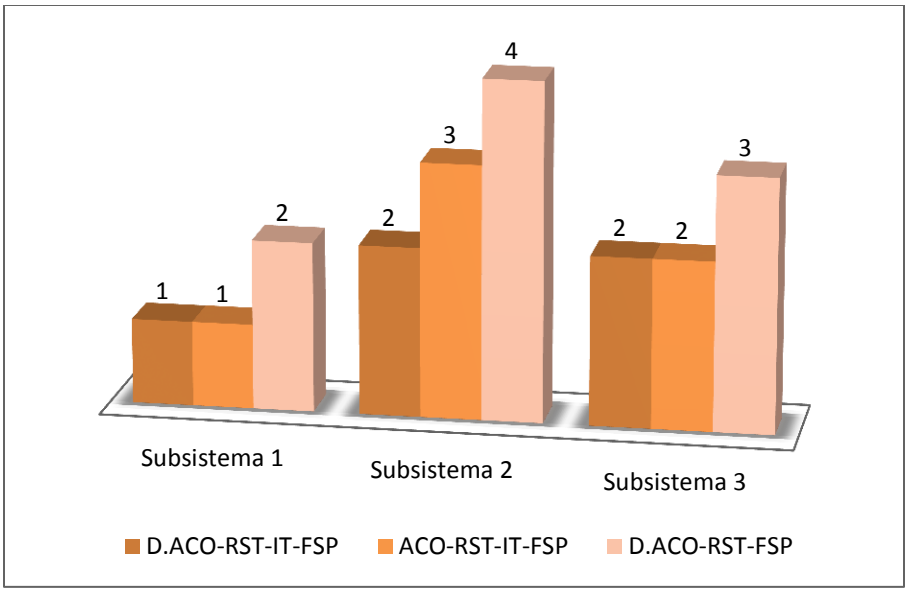

Figura 18 Comparación de los tres modelos para el conjunto de datos Vehicle. (Criterio 3)

Para este conjunto de datos los resultados fueron semejantes a los obtenidos anteriormente para otros datos. El algoritmo propuesto por Gómez Díaz fue mejor en cuanto a la cantidad total de reductos encontrados, sin embargo, en el primer subconjunto de datos este no fue capaz de encontrar el reducto más corto de longitud 14 que encontraron las otras dos variantes.

3.3 Conclusiones del análisis de los resultados

Luego de haber hecho este análisis de los resultados de los algoritmos se puede concluir que en contexto local el algoritmo ACO-RST-IT-FSP obtuvo resultados muy satisfactorios, comparado con otros existentes en la bibliografía y cuya eficacia está probada. En contexto distribuido resultados mostraron que la eficacia del algoritmo propuesto es tan alta como la del modelo D.ACO-RST-FSP, teniendo en cuenta los criterios de calidad asumidos, el cual ha logrado excelentes resultados en los casos que ha sido aplicado.

3.4 Consideraciones parciales

El uso de la medida ganancia de información de la Teoría de la Información puede usarse como medida heurística en un algoritmo poblacional como la optimización basada en colonias de hormigas, lo cual conduce a un algoritmo competente de selección de rasgos.

Esta medida, si se utiliza como metadato de comunicación entre subsistemas, permite extender la aplicabilidad de los algoritmos de selección de rasgos a ambientes con múltiples fuentes de datos.

Si se realiza un estudio de los parámetros, particularmente para los algoritmos propuestos, puede incrementarse la calidad de las soluciones encontradas por estos.

CONCLUSIONES

1. Se desarrolló un nuevo método de selección de rasgos que utiliza conceptos de la Teoría de Conjuntos Aproximados, Teoría de la Información y optimización basada en colonias de hormigas.
2. Se demostró que el uso de la ganancia de información como medida heurística permite construir una función de evaluación de subconjuntos y formar parte de un modelo de selección de rasgos.
3. El método presentado es capaz de buscar reductos satisfactoriamente tanto en contexto distribuido como local.

RECOMENDACIONES

1. Verificar la calidad de los reductos encontrados por ambos modelos propuestos mediante el uso de un clasificador.
2. Realizar un estudio detallado de las configuraciones óptimas de los parámetros de los algoritmos, tanto para la versión distribuida como la no distribuida, para lograr incrementar la calidad de los indicadores de salida.
3. Valorar la combinación de diversas formas de cooperación entre los subsistemas para lograr algoritmos mejor informados de los datos vecinos.

REFERENCIAS

AL-ANI, A. 2005. Feature subset selection using ant colony optimization. *Int. Journal of Computational Intelligence,* 2**,** 53-58.

ALMUALLIM, H. & DIETTERICH, T. Efficient algorithms for identifying relevant features. 9th Canadian Conferece on Artificial Intelligence, 1992. 38-45.

ALMUALLIM, H. & DIETTERICH, T. 1994. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence,* 69**,** 279-305.

BALAMURUGAN, S. A. A. & RAJARAM, R. 2009. Effective and Effcient Feature Selection for Large-scale Data Using Bayes' Theorem. *International Journal of Automation and Computing,* 6**,** 62-71.

BATTITI, R. 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transaction on Neural Networks,* 5**,** 537-550.

BELL, D. & WANG, H. 2000. A formalism for relevance and its application in feature subset selection. . *Machine Learning, ,* 41**,** 175-195.

BELLO, R., NOWÉ, A., CABALLERO, Y., GÓMEZ, Y. & VRANCX, P. Using Ant Colony System meta-heuristic and Rough Set Theory to Feature Selection. The 6th Metaheuristics International Conference (MIC2005), 2005 Vienna, Austria.

BLUM, A. & LANGLEY, P. 1997. Selection of relevant features and examples in machine learning. *Artificial Intelligence,* 97**,** 245-271.

CABALLERO, Y. & BELLO, R. Two new feature selection algorithms with Rough Sets Theory. *In:* BRAMER, M., ed. Artificial Intelligence in Theory and Practice, IFIP 19th World Computer Congress, 2006 Santiago, Chile. Springer Boston, 209-216.

CHOO, Y.-H., BAKAR, A. A. & HAMDAN, A. R. 2008. The fitness-rough: A new attribute reduction method based on statistical and rough set theory. *Intelligent Data Analysis,* 12**,** 73-87.

CLAUSIUS, R. 1865. On Different Forms of the Fundamental Equations of the Mechanical Theory of Heat.

CRAWFORD, B. & CASTRO, C. 2006. Ant Colonies using Arc Consistency Techniques for the Set Partitioning Problem. *LNAI* 4183**,** 45-55.

DÍAZ GALIANO, M. C., MARTÍN VALDIVIA, M. T., MONTEJO RAEZ, A. & UREÑA LÓPEZ, L. A. 2007. Mejora de los sistemas multimodales mediante el uso de ganancia de información.

DOAK, J. 1992. An evaluation of feature selection methods and their application to computer security. *Technical Report CSE-92-18.* Davis, California: University of California, Department of Computer Science.

DOERR, B., NEUMANN, F., SUDHOLT, D. & WITT, C. On the Runtime Analysis of the 1-ANT ACO Algorithm. GECCO 2007, 2007.

DORIGO, M. 1992. *Optimization Learning and Natural Algorithms.* Doctoral Dissertation.

DORIGO, M., BIRATTARI, M. & STUTZLE, T. 2006. Ant colony optimization. *Computational Intelligence,* 1**,** 28-39.

DORIGO, M. & BLUM, C. 2005. Ant colony optimization theory: a survey. *Theory and Computer Science,* 344**,** 243-278.

DORIGO, M., DICARO, G. & GAMBARDELLA, L. M. 1999. Ant colonies for discrete optimization. *Artificial Life,* 5**,** 137-172.

DORIGO, M. & GAMBARDELLA, L. M. 1997. Ant colonies for the travelling salesman problem. *Biosystems,* 43**,** 73-81.

DORIGO, M., MANIEZZO, V. & COLORNI, A. 1996. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems Man, and Cybernetics,* Part B 26**,** 29-41.

DORIGO, M. & STUTZLE, T. 2003. The ant colony optimization metaheuristic algorithms, applications, and advances. *In:* F.GLOVER & KOCHENBERGER, G. A. (eds.) *Handbook of Metaheuristics.* Kluwer.

DORIGO, M. & STUTZLE, T. 2004. *Ant Colony Optimization.,* Cambridge Massachussetts, MIT Press.

DÜNTSCH, I. & GEDIGA, G. 1998. Uncertainty measures of rough set prediction *Artificial Inteligence*.

FIRPI, H. & GOODMAN, E. Swarmed feature selection. Proceedings of the 33rd Applied Imagery Pattern Recognition Workshop (AIPR 2004), 2004. 112-118.

FRAWLEY, W. J., PIATETSKY-SHAPIRO, G. & MATHEUS, C. J. 1992. Knowledge Discovery in Databases: An Overview. *AI Magazine.*

GADAT, S. & YOUNES, L. 2007. A Stochastic Algorithm for Feature Selection in Pattern Recognition. *Journal of Machine Learning Research* 8**,** 509-547.

GAMBARDELLA, L. M., TAILLARD, È. D. & DORIGO, M. 1999. Ant colonies for the Quadratic Assigment Problem. *Journal of the Operational Research Society,* 50**,** 167-176.

GÓMEZ DÍAZ, Y. 2011. *Algoritmos que combinan conjuntos aproximados y optimización basada en colonias de hormigas para la selección de rasgos. Extensión a múltiples fuentes de datos.*, Universidad Central "Marta Abreu" de Las Villas.

HALL, M. Correlation-based feature selection for discrete and numeric class machine learning. 17th International Conference on Machine Learning, 2000. Morgan Kaufmann,

San Francisco, CA, 359-366.

HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P. & WITTEN, I. H. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations,* 11**,** 10-18.

HEINONEN, J. & PETTERSSON, F. 2007. Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem *Applied Mathematics and Computation,* 187**,** 989-998.

HU, X., SHI, J. & WU, X. 2007. A New Algorithm for Attribute Reduction in Decision Tables.

HUANG, K. L. & LIAO, C. J. 2008. Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers and Operations Research,* 35**,** 1030-1046.

JASSO-LUNA, O., SOSA-SOSA, V. & LOPEZ-AREVALO, I. Global Classifier for Confidential Data in Distributed Datasets. *In:* GELBUKH, A. & MORALES, E. F., eds. Mexican International Conference on Artificial Intelligence, 2008. Springer-Verlag Berlin Heidelberg 315-324.

JENSEN, R. 2005. *Combining rough and fuzzy sets for feature selection.* Ph. D, University of Edinburgh.

JENSEN, R. & SHEN, Q. Finding Rough Set Reducts with Ant Colony Optimization. UK Workshop on Computational Intelligence, 2003. 15-22.

JENSEN, R. & SHEN, Q. 2005. Fuzzy-Rough Data Reduction with Ant Colony Optimization. *Fuzzy Set and System.*

K.WANG & SUNDARESH, S. 1998. Selecting features by vertical compactness of data. *Feature extraction, construction and selection.*: Kluwer Academic Publishers.

KIRA, K. & RENDELL, L. A practical approach to feature selection. 9th Int. Conf. on Machine Learning, 1992. 249-256.

KOHAVI, R. & FRASCA, B. Useful Feature Subsets and Rough Set Reducts. Third International Workshop on Rough Sets and Soft Computing, 1994.

KOMOROWSKI, J., PAWLAK, Z., POLKOWSKI, L. & SKOWRON, A. 1999. Rough Sets: A Tutorial. *Rough Fuzzy Hybridization: A new trend in decision making***,** 3-98.

KONONENKO, I. Estimating attributes: Analysis and estensions of relief. European Conference on Machine Learning., 1994 Vienna. 171-182.

KUDO, M. & SKLANSKY, J. 2000. Comparison of algorithms that select features for pattern classifiers. . *Pattern Recognition Letters,* 33**,** 25-41.

LANGLEY, P. Selection of relevant features in machine learning. *Procs. of the AAAI Fall Symposium on Relevance*, 1994. 140-144.

LAZO, M., SHULCLOPER, J. R. & CABRERA, E. A. 2001. An overview of the evolution of the concept of testor. *Pattern Recognition,* 34**,** 753-762.

LIU, H. & MOTODA, H. 2007. *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*, Chapman & Hall/CRC.

LIU, H., MOTODA, H. & DASH, M. A monotonic measure for optimal feature selection. European Conf. on Machine Learning., 1998. Springer Verlag, 101-106.

LIU, H., MOTODA, H., SETIONO, R. & ZHAO, Z. Feature Selection: An Ever Evolving Frontier in Data Mining. *In:* LAWRENCE, N., ed. JMLR: Workshp and Conference Proceedings, 2010.

LIU, H. & SETIONO, R. A probabilistic approach to feature selection: a filter solution. 13th International Conference on Machine Learning., 1996. Morgan Kaufmann, 319-327.

LOWY, J. 2007. Programming WCF Services. *In:* OSBORN, J. (ed.). O'Reilly.

MAGNANI, M. 2003. Technical report on Rough Set Theory for Knowlege Discovery in Data Bases.

MARTENS, D., BACKER, M. D., HAESEN, R., BAESENS, B. & HOLVOET, T. 2006. Ants Constructing Rule-Based Classifiers. *In:* ABRAHAM, A., GROSAN, C. & RAMOS, V. (eds.) *Swarm Intelligence in Data Mining.*

MITCHELL, T. M. 1997. *Machine Learning*, McGraw-Hill.

MUCCIARDE, A. & GOSE, E. 1971. A comparison of seven techniques for choosing subsets of pattern recognition properties. *IEEE Transactions on Computer,* C-20**,** 1023-1031.

NAIMI, H. M. & TAHERINEJAD, N. 2009. New robust and efficient ant colony algorithms: Using new interpretation of local updating process. *Expert Systems with Applications,* 36**,** 481-488.

NARENDRA, P. & FUKUNAGA, K. 1977. A branch and bound algorithm for feature subset selection. . *IEEE Transactions on Computer,* C-26**,** 917-922.

NEUMANN, F. & WITT, C. Runtime analysis of a simple Ant Colony Optimization algorithm. Proceeding of ISAAC '06, 2006. LNCS 4288, 618-627.

NOWÉ, A., VERBEECK, K. & VRANCX, P. Multi-type Ant Colony: The Edge Disjoint Paths Problem. *In:* DORIGO, M., BIRATTARI, M., BLUM, C., GAMBARDELLA, L.M., MONDADA, F., STÜTZLE, T., ed. Ants 2004, 2004 Brussels, Belgium. Springer Verlag, 202-213.

ØHRN, A. 1999. *Discernibility and Rough Sets in Medicine: Tools and Applications.* PhD thesis, Norwegian University of Science and Technology, Department of Computer and Information Science.

ØHRN, A., KOMOROWSKI, J., SKOWRON, A. & SYNAK., P. 1998. The ROSETTA software system. *In:* POLKOWSKI, L. & SKOWRON, A. (eds.) *Rough Sets in Knowledge Discovery 1: Methodology and Applications. Studies in Fuzziness and Soft Computing.* Heidelberg, Germany: Physica-Verlag.

PAWLAK, Z. 1982. Rough sets. *International Journal of Information & Computer Sciences* 11**,** 341-356.

PAWLAK, Z. 1991. *Rough Sets: Theoretical Aspects of Reasoning About Data.,* Dordrecht, Kluwer Academic Publishing.

PAWLAK, Z. & SKOWRON, A. 2007. Rough sets and Boolean reasoning. *Information Sciences,* 177**,** 41-73.

PUDIL, P., NOVOVICOVÁ, J. & KITTLER, J. 1994. Floating search methods in feature selection. *Pattern Recognition Letters,* 15**,** 1119-1125.

QUINLAN, J. R. 1986. Induction of Decision Trees. *Machine Learning*.

RUIZ, R., AGUILAR-RUIZ, J. S. & RIQUELME, J. C. Best Agglomerative Ranked Subset for Feature Selection. *In:* AL., S. E., ed. JMLR: Workshp and Conference Proceedings, 2008.

SHANNON, C. E. 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal*.

SHEINVALD, J., DOM, B. & NIBLACK, W. A modelling approach to feature selection. 10th Int. Conf. on Pattern Recognition, 1990. IEEE Press, 535-539.

SHIE, J.-D. & CHEN, S.-M. 2007. Feature subset selection based on fuzzy entropy measures for handling classification problems.

SIEDLECKI, W. & SKLANSKY, J. 1988. On automatic feature selection. *Int. Journal of Pattern Recognition and Artificial Intelligence.,* 2**,** 197-220.

STÜTZLE, T. & HOOS, H. 2000. MAX-MIN Ant System. *Future Generation Computer Systems,* 16**,** 889-914.

WITTEN, I. H. & FRANK, E. 2005. Data Mining: Practical Machine Learning Tools and Techniques. *Practical Machine Learning Tools and Techniques.* 2 ed.

WOLPERT, D. H. & MACREADY, W. G. 1997. No Free Lunch Theorems for Optimization. *IEEE Transactions of Evolutionary Computation,* 1**,** 67-82.

WONG, K. Y. & SEE, P. C. 2009. A new minimum pheromone threshold strategy (MPTS) for max-min ant system *Journal Applied Soft Computing,* 9**,** 882-888.

WU, Z., ZHAO, N., REN, G. & QUAN, T. 2009. Population declining ant colony optimization algorithm and its applications. *Expert Systems with Applications,* 36**,** 6276-6281.

YAO, Y., ZHAO, Y. & JUEWANG 2006. On Reduct Construction Algorithms.

YU, L. & LIU, H. 2004. Effient feature selection via analysis of relevance and redundancy. *Journal of machine learning research,* 5**,** 1205-1224.

ZHANG, J., WANG, J., LI, D., HE, H. & SUN, J. 2003. A New Heuristic Reduct Algorithm Base on Rough Sets Theory. *WAIM 2003*.

ZHONG, N., DONG, J. & OHSUGA, S. 2001. Using Rough Sets with Heuristics for Feature Selection. *Journal of Intelligent Information Systems,* 16**,** 199-214.

Anexos

Anexo 1 Características de los conjuntos de datos

Conjuntos de Datos.	Número Clases	Número Instancias	Rasgos predictivos		
			Total	Numéricos	Nominales
Dermatology	6	358	34	33	1
LedJen	10	2000	24	0	24
Vehicle	4	846	18	18	0
Breast Cancer W	2	699	8	8	0
HeartJen	2	294	14	0	14
Lung-Cancer	5	32	56	0	56

Anexo 2. Resultados experimentales de la estrategia de ahorro

Breast-Cancer-Wisconsin

Tabla 2 Número de subconjuntos evaluados en cada ciclo

Ciclo (i)	Calculados (ii)	Recuperados de la estructura (iii)	Total de subconjuntos requeridos (iv)
1	126	77	203
2	44	165	209
3	3	199	202
4	6	190	196
5	9	190	199
6	4	196	200
7	8	197	205
8	5	195	200
9	5	199	204
10	6	195	201
11	0	197	197
12	1	204	205
13	1	194	195
14	0	203	203
15	2	200	202

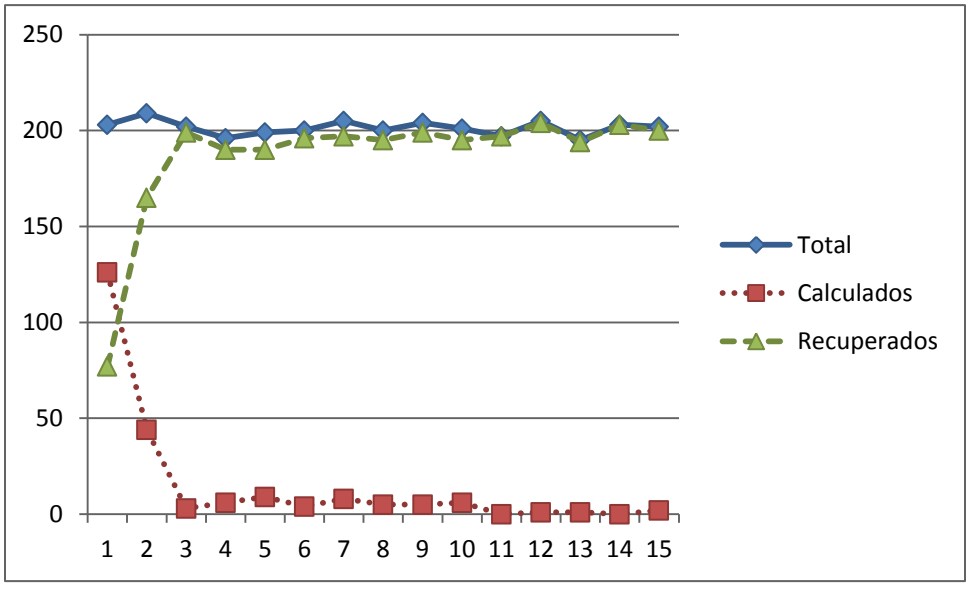

Figura 19 Evaluaciones de subconjuntos

LedJen

Tabla 3 Número de subconjuntos evaluados en cada ciclo

Ciclo (i)	Calculados (ii)	Recuperados de la estructura (iii)	Total de subconjuntos requeridos (iv)
1	6035	346	6381
2	5421	825	6246
3	5149	1027	6176
4	5162	1090	6252
5	4918	1189	6107
6	5087	1235	6322
7	4831	1432	6263
8	4928	1333	6261
9	4694	1572	6266
10	4666	1597	6263
11	4640	1645	6285
12	4598	1690	6288
13	4714	1655	6369
14	4715	1627	6342
15	4679	1595	6274

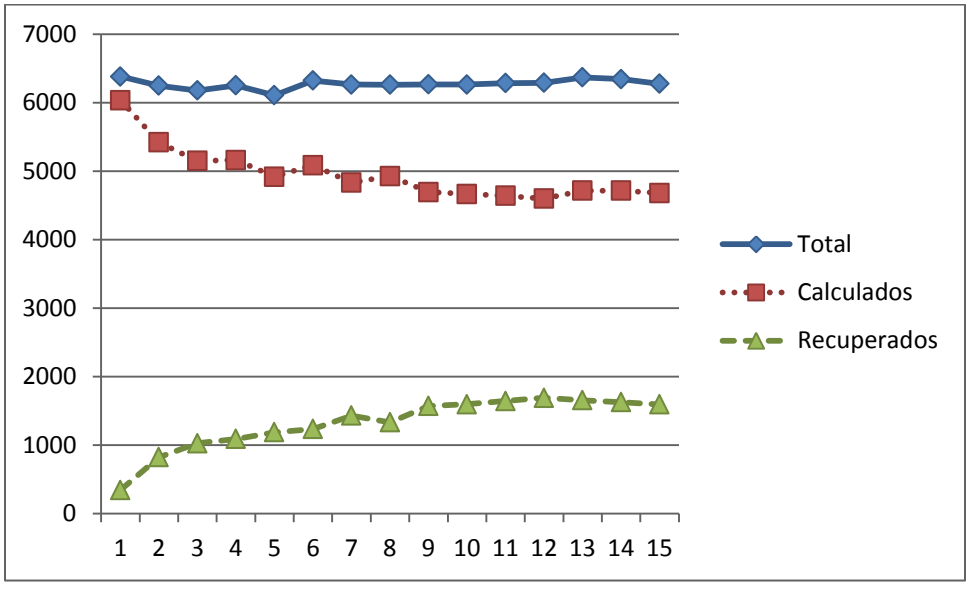

Figura 20 Evaluaciones de subconjuntos

Lung-Cancer

Tabla 4 Número de subconjuntos evaluados en cada ciclo

Ciclo (i)	Calculados (ii)	Recuperados de la estructura (iii)	Total de subconjuntos requeridos (iv)
1	30741	1245	31986
2	30577	3640	34217
3	31926	4494	36420
4	32514	5506	38020
5	30418	5390	35808
6	29363	6518	35881
7	31549	6763	38312
8	29771	7870	37641
9	30623	6500	37123
10	27585	7582	35167
11	27365	8603	35968
12	29426	6985	36411
13	27438	9737	37175
14	29642	7886	37528
15	25927	10461	36388

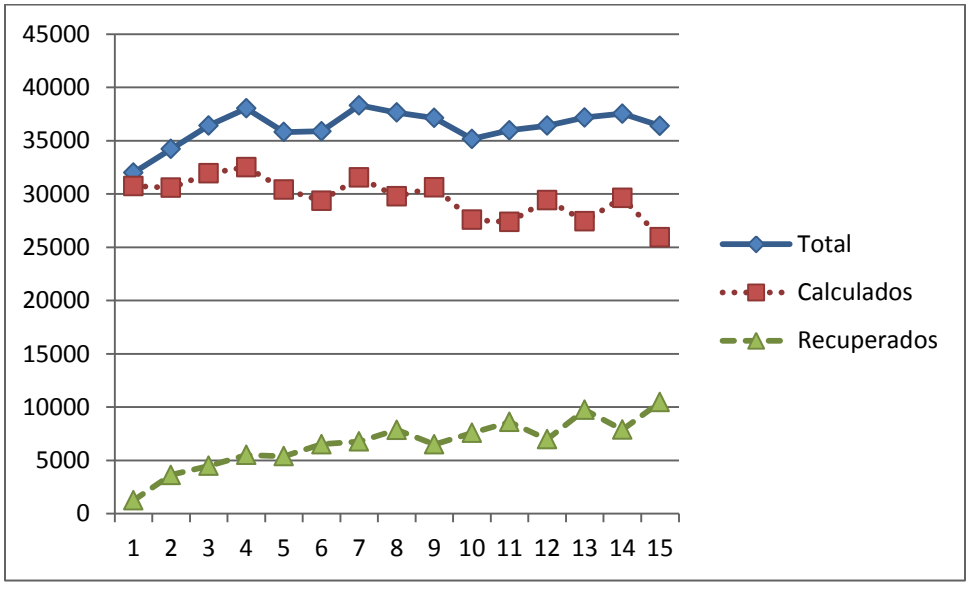

Figura 21 Evaluaciones de subconjuntos

Anexo 3 Comparaciones entre varios algoritmos en contexto local

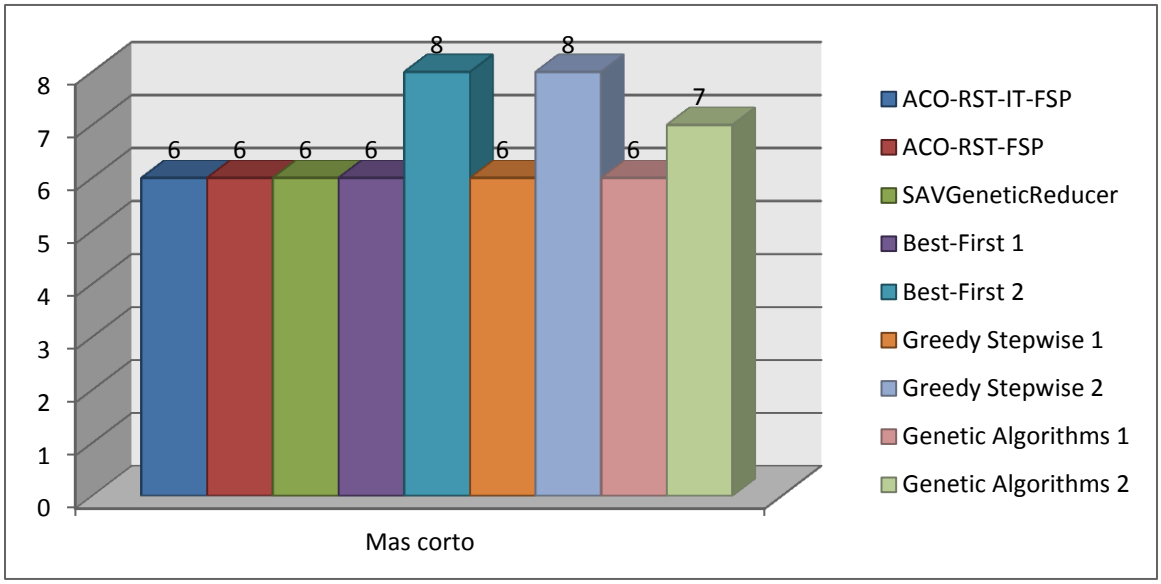

Figura 22 Comparación entre los algoritmos para HeartJen

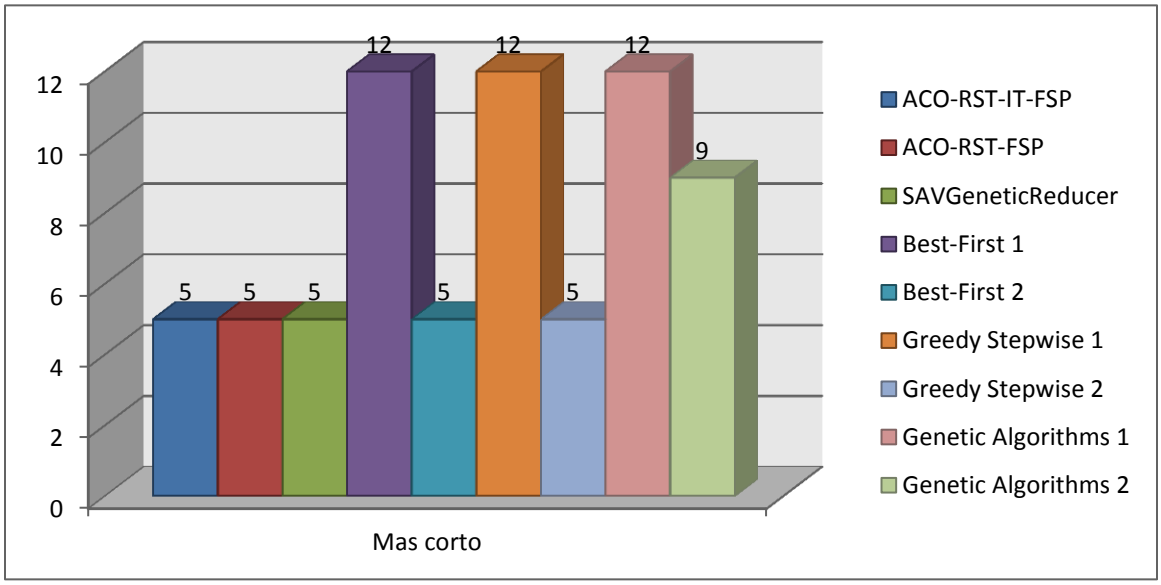

Figura 23 Comparación entre los algoritmos para LedJen