



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**

**FACULTAD DE CIENCIAS DE LA INGENIERÍA**

**Carrera  
INGENIERÍA EN SISTEMAS**

**Tema de la Tesis  
“APRENDIZAJE ESTRUCTURAL DE REDES BAYESIANAS UTILIZANDO  
LA META-HEURÍSTICA OPTIMIZACIÓN BASADA EN MALLAS VARIABLES  
(VMO)”**

**Previo a la obtención del título de:  
INGENIERO EN SISTEMAS**

**Autor:  
LUIS ENRIQUE MOREIRA ZAMORA**

**Director de Tesis  
Phd. Msc. Amilkar Puris Cáceres**

**Quevedo – Ecuador.**

**2015**



# **UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**

## **FACULTAD DE CIENCIAS DE LA INGENIERÍA**

### **Carrera INGENIERÍA EN SISTEMAS**

Presentado al Consejo Directivo como requisito previo a la obtención del título de INGENIERO EN SISTEMAS

Aprobado:

---

ING. WASHINGTON CHIRIBOGA  
PRESIDENTE DEL TRIBUNAL DE TESIS

---

ING. JORGE MURILLO  
MIEMBRO DEL TRIBUNAL DE TESIS

---

ING. CRISTIAN ZAMBRANO  
MIEMBRO DEL TRIBUNAL DE TESIS

---

ING. MARJORIE TORRES  
REDACCIÓN TÉCNICA

## DEDICATORIA

*A mis padres*

*A mi esposa y nuestra hija adorada*

*A mis hermanos*

## **AGRADECIMIENTOS**

*Agradezco a Dios y a todas las personas que han contribuido a la realización de este trabajo:*

*A mis padres Abraham y Bethsy por toda su ayuda, por apoyarme en todo momento, por la comprensión, por los valores que me han inculcado y sobre todo por ser un excelente ejemplo de vida.*

*A mis hermanos y familia por el apoyo brindado siempre.*

*A mi esposa Anara y nuestra adorada hija Itsel, que con su apoyo y dedicación me animan cada día a seguir adelante y no detenerme por nada. Especialmente para ellas que son el motor de mi vida.*

*A toda mi gran familia y a la familia Cáceres Limones.*

*Al Dr. Amilkar Puris Cáceres y familia por permitirme entrar en su hogar como un amigo más y sobre todo por compartir sus sabios conocimientos, su experiencia, su paciencia y su motivación para la realización de este trabajo investigativo.*

*Al Ing. Msc. Byron Oviedo por permitirme ser parte de este magnífico proyecto y sobre todo por el aporte de ideas que fueron uno de los causales principales para encaminar esta investigación.*

*A mis grades e incondicionales amigos, en especial para Cinthya, Irina, Joseph y Carlos, por ayudarme a salir del estrés y por su gran apoyo durante los días de estudio.*

*Para ellos: Muchas gracias y que Dios los bendiga.*

***Luis Enrique Moreira Zamora***

## **DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS**

Yo, LUIS ENRIQUE MOREIRA ZAMORA, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Luis Enrique Moreira Zamora

## CERTIFICACIÓN DEL DIRECTOR DE TESIS

El suscrito, Amilkar Puris Cáceres, Docente de la Universidad Técnica Estatal de Quevedo, certifica que el Egresado LUIS ENRIQUE MOREIRA ZAMORA, realizó la tesis de grado previo a la obtención del título de INGENIERO EN SISTEMAS titulada **“APRENDIZAJE ESTRUCTURAL DE REDES BAYESIANAS UTILIZANDO LA META-HEURÍSTICA OPTIMIZACIÓN BASADA EN MALLAS VARIABLES (VMO)”**, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

---

Phd. Msc. Amilkar Puris Cáceres

## RESUMEN EJECUTIVO

En este trabajo se presenta el estudio de la meta-heurística Optimización Basada en Mallas Variables (VMO) para el entrenamiento estructural de Redes Bayesianas. El algoritmo VMO propone dos operadores principales, expansión y contracción de una malla de soluciones, la cual se encarga de realizar la exploración del espacio de búsqueda. Estos operadores fueron rediseñados para ajustarlos al entrenamiento de redes bayesianas. Para el operador de expansión se estudiaron dos formas de generación de nuevas redes basados en operaciones entre conjuntos y para el caso de la construcción se estudiaron dos formas de selección; elitista y por representatividad.

Los resultados obtenidos luego de realizar un ajuste de los parámetros de VMO se compararon con varios de los clasificadores bayesianos más utilizados en la literatura, donde se obtuvieron resultados significativamente superiores en cuanto a calidad de clasificación.

El algoritmo VMO fue incorporado dentro del software Elvira, el cual fue seleccionado debido a su gran capacidad para el manejo de redes bayesianas y sobre todo por ser de código fuente abierto. Se utilizaron algunas de las clases principales y se crearon nuevas clases que funcionan acorde a las distintas necesidades funcionales del algoritmo.

## **ABSTRACT**

In this work the study of meta-heuristic optimization Meshes Based Variables (VMO) for structural training Bayesian networks is presented. The VMO algorithm proposes two main operators, expansion and contraction of mesh solutions, which is responsible for conducting the exploration of the search space. These operators were redesigned to fit the training of Bayesian networks. Expansion operator two forms new generation networks based operations between sets and for the case of construction selection studied two forms were studied; elitist and representativeness.

The results obtained after adjustment parameters VMO compared with several of Bayesian classifiers commonly used in the literature, which significantly superior results were obtained for quality classification.

The VMO algorithm was incorporated into the, which was selected because of its great capacity for handling Bayesian networks and especially for being open source software Elvira. We used some of the main classes and new classes that operate according to different functional needs of the algorithm are created.

# ÍNDICE

<b>DEDICATORIA</b> .....	<b>iii</b>
<b>AGRADECIMIENTOS</b> .....	<b>iv</b>
<b>DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS</b> .....	<b>v</b>
<b>CERTIFICACIÓN DEL DIRECTOR DE TESIS</b> .....	<b>vi</b>
<b>RESUMEN EJECUTIVO</b> .....	<b>vii</b>
<b>ABSTRACT</b> .....	<b>viii</b>
<b>ÍNDICE</b> .....	<b>ix</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>xiii</b>
<b>ÍNDICE DE ILUSTRACIONES</b> .....	<b>xv</b>
<b>CAPÍTULO I - MARCO CONTEXTUAL DE LA INVESTIGACIÓN</b> .....	<b>1</b>
1.1    Introducción .....	2
1.2    Justificación .....	5
1.3    Problematización .....	6
1.3.1    Formulación .....	7
1.3.2    Sistematización .....	7
1.4    Objetivos.....	8
1.4.1    General .....	8
1.4.2    Específicos.....	8
1.5    Hipótesis.....	9
1.5.1    Planteamiento .....	9
1.5.2    Matriz De Conceptualización.....	9
<b>CAPÍTULO II - MARCO TEÓRICO</b> .....	<b>10</b>
2.1    Fundamentación Conceptual.....	11
2.1.1    Grafo .....	11
2.1.2    Grafo Acíclico.....	11

2.1.3	Redes Bayesianas .....	11
2.1.4	Aprendizaje Automatizado .....	11
2.1.5	Topología de una red .....	11
2.1.6	Algoritmo .....	12
2.1.7	Meta-heurística.....	12
2.2	Fundamentación Teórica .....	12
2.2.1	Redes Bayesianas .....	12
2.2.2	Aprendizaje de Redes Bayesianas.....	13
2.2.3	Aprendizaje Estructural de Redes Bayesianas .....	14
2.2.3.1	Algoritmo para árboles .....	15
2.2.3.2	Algoritmo para poli árboles .....	17
2.2.3.3	Algoritmos para redes múltiplemente conexas .....	19
2.2.4	Los Clasificadores en las Redes Bayesianas.....	19
2.2.4.1	Selección de atributos.....	21
2.2.5	Evaluación de los Clasificadores en las Redes Bayesianas .....	22
2.2.6	Optimización Basada en Mallas Variable .....	24
2.2.6.1	Descripción general de la Meta-heurística .....	25
2.2.6.2	Generación de la malla inicial en cada iteración: .....	26
2.2.6.3	Generación de nodos en dirección a los extremos locales: .....	26
2.2.6.4	Generación de nodos en dirección hacia el extremo global:.....	27
2.2.7	Software para Redes Bayesianas .....	29
2.2.8	Software Elvira .....	29
2.2.9	Técnicas estadísticas utilizadas para el proceso de validación de los resultados .....	31
2.2.9.1	Test de Friedman:.....	32
2.2.9.2	Test de Holm:.....	34
2.3	Marco Referencial .....	37

2.3.1	Algoritmo ByNet .....	37
2.3.2	Algoritmo BayesCHAID .....	38
2.3.3	Algoritmo BayesPSO.....	39
<b>CAPÍTULO III - METODOLOGÍA DE LA INVESTIGACIÓN.....</b>		<b>40</b>
3.1.	Materiales .....	41
3.1.1.	Equipos Y Materiales .....	41
3.1.1.1.	Hardware .....	41
3.1.1.2.	Software.....	41
3.1.1.3.	Suministros .....	42
3.1.1.4.	Personal.....	42
3.1.1.5.	Presupuesto.....	43
3.2.	Métodos y Técnicas Utilizados en la Investigación.....	43
3.2.1.	Tipo de investigación.....	43
3.2.2.	Diseño de Investigación .....	44
3.3.	Población y Muestra .....	45
<b>CAPÍTULO IV - RESULTADOS Y DISCUSIÓN .....</b>		<b>46</b>
4.1	Elementos de la modelación.....	47
4.1.1	Definición del nodo.....	47
4.1.2	Cálculo de la distancia entre nodos.....	48
4.1.3	Generación de nuevas redes a partir de un par .....	48
a)	Unión entre conjuntos .....	48
b)	Unión de las diferencias.....	49
4.1.4	Selección de las mejores redes .....	50
a)	Elitista: .....	50
b)	Representativa:.....	50
4.2	Integración del algoritmo VMO en el software Elvira .....	51

4.3	Aplicación del algoritmo VMO al aprendizaje estructural de Redes Bayesianas.....	53
4.3.1	Generación de la malla inicial en cada iteración .....	53
4.3.2	Generación de nodos en dirección a los extremos locales .....	53
4.3.3	Generación de nodos en dirección a los extremos globales .....	54
4.3.4	Generación de nodos en dirección de frontera .....	54
4.3.5	Contracción de la malla.....	55
4.4	Estudio experimental de VMO y análisis de los resultados .....	55
4.4.1	Análisis de la sección elitista .....	57
4.4.1.1	Operador de Unión .....	58
4.4.1.2	Operador de Unión de las diferencias.....	60
4.4.2	Obtención de la mejor variante para selección elitista .....	61
4.4.3	Análisis de selección representativa .....	62
4.4.3.1	Operador de Unión .....	62
4.4.3.2	Operador de Unión de Diferencias.....	64
4.4.4	Obtención de la mejor variante para selección de Representatividad .....	65
4.4.5	Obtención de la mejor variante para el algoritmo VMO.....	66
4.5	Análisis comparativo de VMO con el estado del arte .....	67
<b>CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>71</b>
5.1	Conclusiones.....	72
5.2	Recomendaciones.....	73
<b>CAPÍTULO VI - BIBLIOGRAFÍA .....</b>		<b>74</b>
6.1	Literatura Citada.....	75
<b>CAPÍTULO VII - ANEXOS.....</b>		<b>80</b>

## ÍNDICE DE TABLAS

Tabla 1. Ejemplo de la matriz de confusión de un problema de dos clases.....	23
Tabla 2. Medidas de evaluación estándar .....	24
Tabla 3. Representación del test iman-davenport.....	35
Tabla 4. Ejemplo de un test de holm para $a_1$ como algoritmo de control .....	35
Tabla 5. Ejemplo del test de wilcoxon .....	36
Tabla 6. Materiales hardware.....	41
Tabla 7. Materiales software .....	41
Tabla 8. Suministros .....	42
Tabla 9. Personal.....	42
Tabla 10. Presupuesto.....	43
Tabla 11. Bases del conocimiento utilizadas .....	45
Tabla 12. Nomenclatura utilizada en las pruebas .....	56
Tabla 13. Test de friedman e iman davenport para $VMO(E_U, k, p)$ .....	58
Tabla 14. Resultado del test de holm para $VMO(E_U, k, p)$ .....	59
Tabla 15. Test de fridman e iman davenport para $VMO(E_D, k, p)$ .....	60
Tabla 16. Resultado del test de wilcoxon selección elitista.....	61
Tabla 17. Test de fridman e iman davenport para $VMO(NE_U, k, p)$ .....	63
Tabla 18. Resultado del test de holm para $VMO(NE_U, k, p)$ .....	64
Tabla 19. Test de fridman e iman davenport para $VMO(NE_D, k, p)$ .....	65

Tabla 20. Resultado del test de wilcoxon selección representativa .....	66
Tabla 21. Resultado del test de wilcoxon para obtener la mejor variante de VMO.....	67
Tabla 22. Test de fridman e iman davenport vmo-estado del arte .....	68
Tabla 23. Resultados del test de holm vmo-estado del arte .....	69
Tabla 24. Resultados obtenidos para VMO(E_U,k,p) .....	81
Tabla 25. Resultados obtenidos para VMO(E_D,k,p) .....	82
Tabla 26. Resultados obtenidos para VMO(NE_U,k,p).....	83
Tabla 27. Resultados obtenidos para VMO(NE_D,K,P).....	84
Tabla 28. Resultados obtenidos para VMO-estado del arte .....	85

## ÍNDICE DE ILUSTRACIONES

Ilustración 1. Definición de un nodo .....	47
Ilustración 2. Ejemplo de la unión de conjuntos .....	49
Ilustración 3. Ejemplo de la unión de diferencias .....	49
Ilustración 4. Valores promedios de las pruebas para $VMO(E_U, k, p)$ .....	59
Ilustración 5. Valores promedios de las pruebas para $VMO(E_D, k, p)$ .....	61
Ilustración 6. Valores promedios de las pruebas para $VMO(NE_U, k, p)$ .....	63
Ilustración 7. Valores promedios de las pruebas para $VMO(NE_D, k, p)$ .....	65
Ilustración 8. Valores promedios de las pruebas VMO-estado del arte .....	69

## **CAPÍTULO I**

### **MARCO CONTEXTUAL DE LA INVESTIGACIÓN**

## 1.1 Introducción

En la actualidad, el constante avance tecnológico exige a las grandes empresas y centros de educación superior, a invertir fuertemente en centros de investigaciones que generen nuevas tecnologías y procesos que nos lleven a la par del crecimiento tecnológico, en este aspecto podemos resaltar una de las ramas fundamentales dentro de este campo como lo es la inteligencia artificial.

La inteligencia artificial es un conjunto de técnicas que nos permiten saber elegir las mejores opciones para resolver algún tipo de problema. Como se menciona en [1]. Según [2] el principal objetivo es obtener conocimiento a través de los datos, utilizando modelos de adaptación o procesos de inferencia. El enfoque de probabilidad bayesiana está considerado como una arquitectura agrupada que se encuentra dentro de los métodos de entrenamiento automatizado para la modelación e inferencia.

El enfoque probabilístico bayesiano o mejor conocido como Redes Bayesianas es una técnica de inteligencia artificial que ha mostrado excelentes resultados en el manejo de grandes volúmenes de datos, debido a que representan las relaciones que existen entre las variables y en base a su caracterización se seleccionan las más idóneas que represente la mejor solución.

El obtener una red bayesiana a partir de datos se denomina proceso de aprendizaje y consiste en dos subprocesos diferentes entre sí [3]:

- **Aprendizaje Estructural:** Este tipo de aprendizaje permite obtener las relaciones de dependencia entre las variables del modelo a partir de un conjunto de datos observados, determinando la topología del grafo que mejor describe el sistema que se está analizando. El resultado de este tipo de aprendizaje es la obtención de los arcos que unen los diferentes nodos de la red.

- **Aprendizaje paramétrico:** Este tipo de aprendizaje permite calcular los valores de los parámetros o valores de las filas de las tablas de probabilidad condicionada, a partir de la topología del grafo, el tipo de red Bayesiana y el conjunto de datos observados.

El aprendizaje estructural se define como un problema de optimización discreto ya que el objetivo es encontrar de todas las topologías posibles, aquella que mejor represente las relaciones de las variables, por este motivo se han venido desarrollando diferentes propuestas para resolver problemas de aprendizaje estructural. En [2] se pueden ver diferentes alternativas de Meta-heurística para darle solución aproximada a este problema, donde se ha podido evidenciar que no existe todavía el modelo que plantee la mejor solución.

Por otro lado, aparece en la literatura una Meta-heurística recientemente introducida en la comunidad científica, la Optimización basada en Mallas Variables (VMO) [4] como un modelo que ha presentado soluciones muy buenas en las cuales por su funcionamiento y su característica evolutiva propone una nueva forma de exploración del espacio de búsqueda.

Es por este motivo que el presente trabajo está encaminado a encontrar una forma de aplicar la optimización de Mallas Variables al problema discreto de entrenamiento de redes bayesianas ya que aún no se presenta ninguna solución que obtenga los mejores resultados para cualquier base del conocimiento. Para la ejecución del siguiente trabajo se utilizarán bases del conocimiento internacionales que se encuentran en UC Irvine Machine Learning Repository [5].

Para el desarrollo de este trabajo se hace uso de un software llamado Elvira de referencia internacional, desarrollado en la comunidad científica española mediante un proyecto coordinado con algunas universidades, el cual está destinado a la edición y evaluación de modelos gráficos probabilísticos, concretamente redes bayesianas y diagramas de influencia.

Así mismo se utilizará un conjunto de algoritmos de entrenamiento estructural ya desarrollado como ByNet, BayesChaid, BayesPSO, RB K2, RB TAN y CBN que se encuentran en [2] así como también Naïve Bayes y Naïve Bayes aumentado a árbol (TAN) que viene incorporado en el software Elvira, que servirán como base de comparación con nuestra propuesta.

## 1.2 Justificación

El aprendizaje estructural de una red bayesiana representa uno de los elementos fundamentales en el desempeño de un clasificador bayesiano. Este problema cae en la categoría NP-Completo [6], por lo que ha sido objeto de estudio de un sin número de investigaciones en el área. Dentro de los algoritmos aproximados, las meta heurísticas poblacionales son las que más han sido utilizadas mejorando en gran medida el rendimiento de los clasificadores, aunque aún no se ha encontrado el mejor de los algoritmos de entrenamiento para cualquier problema de estudio.

Por otra parte los clasificadores bayesianos han sido aplicados con éxito en numerosos estudios, sin importar el campo en que se lo aplique. Esto se debe a la facilidad que posee para el modelamiento de la incertidumbre de sistemas expertos y para resolver los problemas comunes que tiene el proceso de clasificación, inferencia, etc.

De lo antes expuesto se deduce la necesidad de presentar nuevos algoritmos para el aprendizaje automatizado de redes bayesianas con el objetivo de obtener topologías que represente de manera más conveniente las relaciones de dependencia entre variables y de esta forma mejorar la calidad de los clasificadores bayesianos.

Conjuntamente con lo antes mencionado este trabajo se enmarca dentro de un proyecto FOCICYT aprobado en la facultad sobre “ALGORITMOS META-HEURÍSTICOS PARA EL APRENDIZAJE DE REDES BAYESIANAS” que tiene como fin encontrar un modelo teórico para el aprendizaje bayesiano que permita estudiar la deserción estudiantil en la universidad.

### 1.3 Problematización

El proceso de aprendizaje estructural se fundamenta en la utilización de una estrategia de búsqueda para explorar todas las posibles topologías de la red Bayesiana. Según [2] el interés de la Inteligencia Artificial en el aprendizaje de Redes Bayesianas, se fundamenta en la complicitad entre la incertidumbre y el entrenamiento automatizado.

El aprendizaje estructural define las relaciones de dependencia entre las variables, de forma que se pueda determinar la topología de la red bayesiana que mejor describa el sistema que se está analizando, es por esto que nuestro estudio se llevará a cabo para definir otra forma de aprendizaje estructural basado en la búsqueda de estructuras a través de una meta-heurística poblacional [2].

Una de las primeras ideas en este sentido fue la propuesta de un algoritmo genético con codificación binaria presentado en [7]. Otro algoritmo evolutivo utilizado fue la programación evolutiva [8] que utiliza la misma codificación del algoritmo genético anterior, pero trabaja con cinco operadores para la generación de hijos y recurre a un espacio reducido utilizando el test de independencia condicional [2].

Una Meta-heurística recientemente introducida en la comunidad científica es la Optimización basada en Malla Variables [4] es una Meta-heurística poblacional con características evolutivas que propone una nueva forma de exploración del espacio de búsqueda probado en dominios reales con muy buenos resultados. De aquí la necesidad por probar su desempeño en el problema del aprendizaje estructural de redes bayesianas, para ello debemos proponer una estructura del algoritmo para ambientes discretos.

Actualmente la Universidad Técnica Estatal de Quevedo a través de la FOCICYT se encuentra desarrollando una investigación sobre el entrenamiento de redes bayesianas para ejecutar medidas preventivas que permitan reducir el

índice de deserción académica, para lo cual utilizan el sistema Elvira, este programa cuenta con las siguientes prestaciones: interfaz gráfica para la construcción de redes bayesianas, algoritmo de toma de decisiones, aprendizaje de modelos a partir de bases de datos, fusión de redes, etc.

### **1.3.1 Formulación**

¿Cómo modelar la Meta-heurística Optimización Basada en Mallas Variables para dar solución al problema discreto de entrenamiento de redes bayesianas?

### **1.3.2 Sistematización**

- ¿Cómo redefinir los operadores de exploración y contracción de la Meta-heurística Optimización basada en Mallas Variables para el problema discreto de estudio?
- ¿Cómo Realizar un estudio de parámetros satisfactorio de manera que se pueda obtener la mejor de las variantes implementadas?
- ¿De qué manera se puede validar la calidad de los resultados que se obtendrán de la Meta-heurística Optimización Basada en Mallas Variables?

## **1.4 Objetivos**

### **1.4.1 General**

Obtener un modelo basado en Optimización basada en Mallas Variables para entrenamiento estructural de redes bayesianas.

### **1.4.2 Específicos**

- Definir los operadores de exploración y contracción de la meta-heurística.
- Realizar un estudio de parámetros indicando cuál de las posibles variantes sea la de mejor comportamiento.
- Comparar los resultados alcanzados por la propuesta con otros algoritmos propuestos en documentos citados.

## 1.5 Hipótesis

### 1.5.1 Planteamiento

¿La Optimización Basada en Mallas Variables obtendrá resultados competitivos en el entrenamiento estructural de redes bayesianas utilizando las bases del conocimiento de UCI Repository?

### 1.5.2 Matriz De Conceptualización

Variables	Definición Conceptual	Dimensiones	Indicadores
<b>INDEPENDIENTE</b>			
Algoritmo de optimización basado en mallas variables	Meta-heurística poblacional de exploración del espacio de búsqueda complejo.	Desplazamiento	<ul style="list-style-type: none"><li>• Exploración del espacio de búsqueda complejo.</li></ul>
<b>DEPENDIENTE</b>			
Aprendizaje estructural de redes bayesianas	Mecanismo de entrenamiento de redes bayesianas por el cual se obtiene una red a partir de una base de datos experimental de la UCI Repository.	Eficacia	<ul style="list-style-type: none"><li>• Generar una topología de redes bayesianas lo más acorde al problema.</li></ul>

## **CAPÍTULO II**

### **MARCO TEÓRICO**

## **2.1 Fundamentación Conceptual**

### **2.1.1 Grafo**

Según [9] es una forma de representar gráficamente diversos puntos a los cuales se los denomina nodos, estos nodos se localizan unidos por medio de líneas a las cuales se las denominan aristas. Esto sirve para visualizar la formación de relaciones recíprocas.

### **2.1.2 Grafo Acíclico**

Según [10] un grafo dirigido acíclico es un grafo que no contiene ciclos, es decir contiene aristas que están identificadas por un camino simple que no permite repetir vértices.

### **2.1.3 Redes Bayesianas**

Según [11] Una red bayesiana es considerado como un grafo acíclico dirigido acíclico en el que cada variable es representada por un nodo y la dependencia probabilística es simbolizada por aristas, creando una relación de causa-efecto entre el nodo padre y el nodo hijo.

### **2.1.4 Aprendizaje Automatizado**

El aprendizaje puede ser definido como “cualquier proceso a través del cual un sistema mejora su eficiencia”. La habilidad de aprender es considerada como una característica central de los “sistemas inteligentes” [12] [13].

### **2.1.5 Topología de una red**

La topología o estructura de la red nos da información sobre las dependencias probabilísticas entre las variables pero también sobre las independencias condicionales de una variable (o conjunto de variables) dada otra u otras variables; dichas independencias simplifican la representación del conocimiento

(menos parámetros) y el razonamiento (propagación de las probabilidades) [11].

### **2.1.6 Algoritmo**

En matemáticas, lógica, ciencias de la computación y disciplinas relacionadas, un algoritmo (del griego y latín, dixit algorithmus) es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad [14].

### **2.1.7 Meta-heurística**

Una Meta-heurística es un método heurístico para resolver un tipo de problema computacional general, usando los parámetros dados por el usuario sobre unos procedimientos genéricos y abstractos de una manera que se espera eficiente. Normalmente, estos procedimientos son heurísticos [14].

## **2.2 Fundamentación Teórica**

### **2.2.1 Redes Bayesianas**

Dentro de la inteligencia artificial podemos citar a las redes bayesianas como uno de los más importantes modelos probabilísticos, debido a su eficacia al modelar la incertidumbre en el mundo real.

Según [2] son importantes debido a que simulan las relaciones de dependencia e independencia que contienen las variables entre sí, esto se debe a la estructura propia de las redes bayesianas que le permiten representar conocimiento de forma reducida ya que utiliza una menor cantidad de parámetros en consideración con otros modelos probabilísticos.

Según [2]: Una red bayesiana se define como un grafo que cumple lo siguiente (Stuart, 1996, 2003):

- Cada nodo de la red está compuesto por variables aleatorias que se representan mediante la letra X o utilizando subíndices  $X_1, X_2, X_3 \dots X_n$ . Pueden ocurrir casos en que la variable original tenga que descomponerse en variables aleatorias, esto procede cuando existen múltiples valores para una sola variable, para el resto de los casos se representan como atributos
- Debe existir una relación con cada par de nodos utilizando arcos dirigidos, esto nos permite saber que una variable ejerce una dependencia condicional hacia la otra variable.
- Se debe contar con una tabla de probabilidad condicional que nos permitirá ponderar la influencia de los nodos padres sobre los nodos hijos.
- No puede contener ciclos dirigidos, es por esto que una red bayesiana es considerada como un grafo dirigido acíclico.

### **2.2.2 Aprendizaje de Redes Bayesianas**

El aprendizaje es una de las características que definen a los sistemas basados en inteligencia artificial porque siendo estrictos se puede afirmar que sin aprendizaje no hay inteligencia; es difícil definir el término “aprendizaje”, pero la mayoría de las autoridades en el campo coinciden en que es una de las características de los sistemas adaptativos que son capaces de mejorar su comportamiento en función de su experiencia pasada, por ejemplo al resolver problemas similares [15].

El aprendizaje en la redes bayesianas consiste en definir la red probabilística a partir de datos almacenados en bases de datos en lugar de obtener el conocimiento del experto. Este tipo de aprendizaje ofrece la posibilidad de inducir la estructura gráfica de la red a partir de los datos observados y de definir las relaciones entre los nodos basándose también en dichos casos; según [3] a estas dos fases se las puede denominar respectivamente aprendizaje estructural y aprendizaje paramétrico, los cuales se detallan a continuación.

- **Aprendizaje estructural:** el objetivo principal de esta fase es obtener una estructura de red bayesiana partiendo de bases de datos, es decir, se toma en consideración las relaciones de dependencia e independencia entre todas las variables. Las diferentes técnicas de aprendizaje estructural basan su procesamiento en el tipo de estructura o topología de la red (árboles, poli-árboles o redes múltiplemente conectadas). Según [2] podemos considerar la opción de congregarse el conocimiento de expertos con este tipo de aprendizaje para obtener una estructura más exacta en base al mundo real.
- **Aprendizaje paramétrico:** dada una estructura y las bases de datos, obtiene las probabilidades a priori y condicionales requeridas [2].

### 2.2.3 Aprendizaje Estructural de Redes Bayesianas

Según [2] el modelo bayesiano Naïve o Naïve-Bayes presentado por [16] se ha convertido en un referente con el cual comparar los beneficios de diferentes técnicas debido a su fácil utilización y que es uno de los modelos más simples.

Es considerado un modelo simple debido a que las variables predictivas son limitadamente independientes dada la variable a clase, esto quiere decir que queda definida una estructura y que lo único que resta por hacer es aprender las probabilidades en base a los valores de los atributos de la variable clase.

Una opción para optimizar la estructura de un modelo Naïve-Bayes sería añadiendo arcos entre nodos que tengan dependencia. Bajo este concepto se han logrado implementar variaciones del modelo Naïve-Bayes como [17] y [18].

Aunque una muy buena variante sería aplicando las siguientes operaciones locales hasta conseguir el objetivo propuesto:

- La eliminación de atributos.
- La unión de atributos contenidos en una variable combinada.
- Añadir nuevos atributos en base a dos atributos dependientes, logrando independencia entre los mismos.

Según [2] este objetivo se puede lograr probando todas las opciones antes expuestas, midiendo la dependencia entre los atributos tomando en consideración la variable clase o a su vez la información mutua condicional utilizando la siguiente expresión:

$$IMC(X_t, X_j|C) = \sum_{X_t, X_j} P(X_t, X_j|C) \frac{\log(P(X_t, X_j|C))}{P(X_t|C)P(X_j|C)} \quad (2.1)$$

### 2.2.3.1 Algoritmo para árboles

Según [2] el entrenamiento estructural para árboles se fundamenta en el algoritmo presentado por [19] para acercar la distribución de probabilidades hacia un resultado que nos arroje probabilidades de segundo orden:

Se establece desde un problema de optimización con el objetivo de obtener una estructura en forma de árbol, la misma que se aproxime al mundo real. Para lograr este objetivo se manipula la diferencia de información entre la distribución real (P) y la aproximada (P\*) usando la siguiente expresión:

$$D(P, P^*) = \sum_X P(X) \log (P(X)/P^*(X)) \quad (2.2)$$

El fin a seguir sería minimizar D, para lograr este objetivo se define la diferencia en base a la información mutua entre dos variables, con la siguiente expresión:

$$D(X_t, X_j) = \sum_X P(X_t, X_j) \frac{\log (P(X_t, X_j))}{P(X_t)P(X_j)} \quad (2.3)$$

Con esto se logra observar que la diferencia de información se basa en la suma de la información mutua de todos los nodos pares que conforman el árbol utilizando signo cambiado, por ello se desea conseguir un árbol con mayor peso.

Según [2] los pasos del algoritmo que se debe utilizar para determinar el árbol bayesiano óptimo a partir de base de datos es el que se detalla a continuación:

1. Se computariza la cantidad de información en común que tienen todos los pares de nodos  $(n(n-1)/2)$ .
2. Se genera una lista en orden descendente de los resultados obtenidos en el paso anterior.
3. Se selecciona la rama de alto valor como árbol inicial.
4. A continuación se agrega la siguiente rama considerando la formación de ciclos, en el caso de que la siguiente rama forme un ciclo se la desecha.
5. Se repite el paso anterior hasta que se logren cubrir todas los nodos  $(n-1$  ramas).

Se puede utilizar la opinión de algún experto debido a que el algoritmo no facilita la direccionalidad de las aristas, por lo que fácilmente se puede determinar de forma arbitraria.

Otro problema a considerar es la búsqueda incompleta realizada por el algoritmo que ocasiona problemas de sobre ajuste, es por esto que se debe utilizar el estadístico Chi-cuadrado.

Este algoritmo no tiene la ejecución vuelta atrás o backtracking, una vez que el algoritmo haya escogido un nodo, jamás volverá a considerar esa selección. Por lo que se vuelve susceptible a los mismos riesgos de colinas.

### **2.2.3.2 Algoritmo para poli árboles**

Según [2] en [20] se presenta el algoritmo de [19] en una versión más extendida para poli árboles. Para este objetivo se parte de una estructura sin direcciones obtenida con el algoritmo anterior y se establece la dirección de las aristas utilizando pruebas de independencia entre grupos de variables de tres miembros.

Con este proceso se obtiene una estructura de red bayesiana en forma de poli árbol, El algoritmo presentado en [20] se basa en realizar pruebas de relación de dependencia entre todos los grupos de variables que se encuentran en el esqueleto. Como se tienen grupos de tres variables, solo pueden existir tres casos:

1. Arcos divergentes:  $X \leftarrow Y \rightarrow Z$
2. Arcos secuenciales:  $X \rightarrow Y \rightarrow Z$
3. Arcos convergentes:  $X \rightarrow Y \leftarrow Z$

Dentro de los casos posibles, los dos primeros son indistinguibles, pero el último se considera diferente debido a que los nodos padre son independientes. Dentro de los pasos del algoritmo podemos encontrar los siguientes:

1. Se obtiene la estructura en tomando como base el resultado del algoritmo de [19].
2. Se realiza una búsqueda sobre la red para encontrar un grupo de nodos de tres miembros que sean convergentes o tengan nodos multi padre.
3. Se parte de un nodo multi padre para determinar en qué dirección irán los arcos, para este caso se utiliza la prueba de grupo.
4. Repetir los dos pasos anteriores hasta cubrir la cantidad máxima de arcos.
5. En el caso de que queden arcos sin direccionar, se utiliza la opinión del experto para determinar la misma.

Este algoritmo tiene como principal restricción el hecho de que está configurado para poli árboles, esto no garantiza obtener el máximo de direcciones permitidas.

Un problema muy común dentro de este algoritmo es el hecho que por lo general no se establece la independencia absoluta, por lo que se consideran métodos empíricos para poder aproximar esa independencia.

### **2.2.3.3 Algoritmos para redes múltiplemente conexas**

Según [2] se pueden apreciar dos tipos de métodos dentro del aprendizaje de redes bayesianas, dentro de los cuales se incluyen las redes múltiplemente conexas, las cuales se detallan a continuación:

1. Métodos que basan su funcionamiento en medidas de búsqueda y ajuste.
2. Métodos que basan su funcionamiento en pruebas de independencia o también logran basarse en restricciones.

El primer tipo de métodos tiene como aspectos principales: el mejoramiento de la medida para evaluar en cada estructura respecto a los datos y una técnica de búsqueda para generar múltiples estructuras hasta lograr obtener la mejor, tomando en consideración la medida seleccionada.

Dentro de este tipo de métodos encontramos una ventaja muy importante con respecto a los métodos que se basan en restricciones, puesto que se apoyan en encontrar un modelo basado en la información en categorías de la independencia condicional entre los nodos.

### **2.2.4 Los Clasificadores en las Redes Bayesianas**

Según [2] se puede definir a un problema de clasificación partiendo de un conjunto de entrenamiento ( $T$ ) y un grupo de clases ( $C$ ), pudiendo encontrar la clase  $C_j$  de un objeto  $t_i$  sin utilizar  $T$ .

Se puede utilizar como clasificador una red bayesiana solo cuando exista un nodo no evidenciado que precisamente representa la variable clase, en esta circunstancia se habla de un clasificador bayesiano. Utilizando la siguiente

función, los clasificadores bayesianos logran disminuir el costo del error durante la clasificación:

$$\gamma(x) = \arg \min_k \sum_{c=1}^{r_0} \text{cost}(k, c) p(c|x_1, x_2, \dots, x_n)$$

Donde  $\text{cost}(k, c)$  expresa el valor de una clasificación mala según algunos criterios. Cuando existen más de una variable clase, se tiende a tomar en consideración la variable más a priori como se expresa en la siguiente función:

$$\gamma(x) = \arg \max_x p(c|x_1, x_2, \dots, x_n) = \arg \max_c p(C)p(x_1, x_2, \dots, x_n|C)$$

Según [2] se pueden obtener algunos clasificadores como los mostrados en [7] dependiendo de la manera que se pueda aproximar  $p(X_1, X_2, \dots, X_N | C)$ .

Cuando tenemos el caso de que la variable clase forma parte del conjunto de datos de entrenamiento, es cuando podemos decir que las redes bayesianas son usadas como clasificadores, como lo dice [18].

Podemos citar el clasificador Naïve Bayes como uno de los más usados, se aplica teorema de bayes para calcular la probabilidad de que el  $i$ -ésimo ejemplo pertenezca a la clase  $j$ -ésima, este clasificador no obtiene resultados favorables en casos donde las variables se encuentren correlacionadas.

En [21] se presenta una versión mejorada del modelo clasificador Naïve Bayes, al cual denominaron Naïve Bayes aumentado a árbol (TAN por sus siglas en ingles), este método obtiene mejores resultados que el método tradicional manteniendo la simplicidad computacional.

Básicamente su funcionamiento se basa en la ideología de que la variable clase no tiene padres y tiene que ser padre de todas las variables predictoras,

cabe destacar que este algoritmo es una adaptación del algoritmo de [19] y calcula la información que tienen en común un par de nodos. Para tener una idea más profunda del funcionamiento de este algoritmo, se puede encontrar en [22] una detallada descripción de su funcionamiento.

Por otro lado tenemos el k Dependence Bayesian classifier (kDB, clasificador bayesiano con k dependencias) presentado por [23]. Este algoritmo tiene la ventaja de evitar la restricción que tiene el TAN puesto que una variable predictora puede tener k número de padres.

#### **2.2.4.1 Selección de atributos**

Según [2] dentro del proceso de aprendizaje el problema más común se presenta al momento de seleccionar los atributos para la descripción de los datos. Nos podemos encontrar con rasgos innecesarios para el proceso de aprendizaje y la mayoría de algoritmos no cumplen bien su función debido a este problema. Por ello se hace presente la necesidad de la utilización de técnicas que te faciliten el proceso de reconocer los atributos innecesarios para el proceso de aprendizaje.

Según [2] actualmente se pueden apreciar dos aproximaciones para tratar de aplacar este problema:

- Se utilizan las técnicas no supervisadas que nos permite reducir el espacio de dimensiones tal como el análisis de componentes principales presentado en [24], [25], [26], escalado multidimensional presentado en [27] o proyección aleatoria presentado en [28].
- Se utilizan las técnicas supervisadas o técnicas de filtra para establecer un conjunto de atributos más adecuado, en este aspecto la más utilizada es la técnica de envoltura presentada por [29].

**Técnicas de filtrado:** Esta técnica se basa en la medida de evaluación que contiene cada uno de los atributos, esto permite saber el nivel de relevancia en comparación con el objetivo propuesto, esto se vuelve computacionalmente de bajo costo debido a que el siguiente proceso es ordenar en forma descendente los atributos e ir eliminando los atributos innecesarios.

**Técnicas de envoltura:** Esta técnica a diferencia de la anterior se vuelve computacionalmente más costosa debido a que prueban todos los subconjuntos de atributos para encontrar el más óptimo, para este proceso de prueba no se puede hacer una búsqueda totalmente exhaustiva y por lo general se hace necesaria la utilización de algún algoritmo de aprendizaje.

Dentro de esta técnica podemos encontrar la de ascensión de colinas o Hill-climbing, la de recocido simulado o simulated annealing, la de primero el mejor o best firsts, etc. Básicamente se pueden utilizar dos estrategias muy comunes, ya sea la de selección hacia adelante o la eliminación hacia atrás.

## **2.2.5 Evaluación de los Clasificadores en las Redes Bayesianas**

Según [2] como en todo proceso de aprendizaje, cuando se hace necesaria la utilización de redes bayesianas como clasificadores, se debe realizar seguidamente un proceso de evaluación de desempeño. Para este proceso se llevan a cabo criterios como el porcentaje de clasificaciones correctas, medidas de error, el índice Kappa presentado en [30], la medida F observado en [31], y funcionales de calidad y error estudiado en [32] y [33].

La precisión está ligada en su totalidad a la capacidad que tiene un modelo para personificar confiablemente un problema del mundo real, cabe destacar que no existe un modelo concreto que obtenga resultados óptimos para problemas generales, es por esto que se ha visto la necesidad de estudiar

varias medidas para aplicar la evaluación del clasificador y poder comparar con modelos utilizados en ese mismo problema.

Dentro de este campo podemos encontrar a la matriz de confusión como la medida más conocida al momento de evaluar un modelo de clasificación, esta matriz se obtiene al probar el resultado del clasificador con un conjunto de datos de entrenamiento o un conjunto de datos externos.

En la Tabla 1 se puede apreciar la estructura de la matriz de confusión para un problema simple que contiene dos clases, donde Pos/pos representa a una clase positiva y Neg/neg una clase negativa.

Tabla 1. Ejemplo de la matriz de confusión de un problema de dos clases

Matriz de Confusión		Clase Verdadera	
		Pos	Neg
Clase Predicha	Pos	VP	FP
	Neg	FN	VN
Total Columna		P	N

En la Tabla 1 las siglas VP y VN simbolizan elementos que han sido adecuadamente clasificados de la clase positiva y negativa. FP y FN representan elementos que han tenido una mala clasificación. Basándonos en estos datos se puede calcular el error, la exactitud, la sensibilidad, la precisión y la especificidad, el coeficiente de correlación de Matthews (mcc), que se muestran en las expresiones de la Tabla 2 [2].

Tabla 2. Medidas de evaluación estándar

Nombre	Medida
Exactitud	$\frac{VP + VN}{P + N}$
rVP o sensibilidad	$\frac{VP}{P}$
rVN o especificidad	$\frac{VN}{N}$
rFP	$\frac{FP}{N}$
rFN	$\frac{FN}{P}$
Precisión	$\frac{VP}{VP + FP}$
Medida F	$\frac{2}{\frac{1}{\text{precisión}} + \frac{1}{\text{sensibilidad}}}$
Correlacion de Matthews (mcc)	$\frac{VP * VN - FP * FN}{\sqrt{(VP + FN)(VN + FP)(VP + FP)(VN + FN)}}$

### 2.2.6 Optimización Basada en Mallas Variable

La Optimización Basada en Mallas Variable (Variable Mesh Optimization; VMO) [34] es una Meta-heurística poblacional con características evolutivas donde un conjunto de nodos que representan soluciones potenciales a un problema de optimización, forman una malla (población) que dinámicamente crece y se desplaza por el espacio de búsqueda (evoluciona). Para ello, se realiza un proceso de expansión en cada ciclo, donde se generan nuevos nodos en dirección a los extremos locales (nodos de la malla con mejor calidad en distintas vecindades) y el extremo global (nodo obtenido de mejor calidad en todo el proceso desarrollado); así como a partir de los nodos fronteras de la malla. Luego se realiza un proceso de contracción de la malla, donde los mejores nodos resultantes en cada iteración son seleccionados como malla

inicial para la iteración siguiente. La formulación general de la Meta-heurística abarca tanto los problemas de optimización continuos como los discretos.

### **2.2.6.1 Descripción general de la Meta-heurística**

La esencia del método VMO es crear una malla de puntos en el espacio  $m$  dimensional, donde se realiza el proceso de optimización de una función  $FO(x_1, x_2, \dots, x_m)$ ; la cual se mueve mediante un proceso de expansión hacia otras regiones del espacio de búsqueda.

Dicha malla se hace más “fina” en aquellas zonas que parecen ser más promisorias. Es dinámica en el sentido que la malla cambia su tamaño (cantidad de nodos) y configuración durante el proceso de búsqueda. Los nodos se representan como vectores de la forma  $n(x_1, x_2, \dots, x_m)$ .

El proceso de generación de nodos en cada ciclo comprende los pasos siguientes:

1. Generación de la malla inicial.
2. Generación de nodos en dirección a los extremos locales ( $n_l$ ).
3. Generación de nodos en dirección al extremo global ( $n_g$ ).
4. Generación de nodos a partir de las fronteras de la malla ( $n_f$ ).

El método incluye los parámetros:

1. Cantidad de nodos de la malla inicial ( $N_i$ ).
2. Cantidad máxima de nodos de la malla en cada ciclo ( $N$ ), donde  $3 \cdot N_i \leq N$ .

3. Tamaño de la vecindad ( $k$ ).
4. Condición de parada ( $M$ ).

A continuación se presenta una descripción más detallada de cada uno de los pasos de VMO en el proceso de generación de nuevos nodos (expansión):

#### **2.2.6.2 Generación de la malla inicial en cada iteración:**

La malla inicial consta de  $N_i$  nodos, los cuales en la primera iteración son generados de forma aleatoria o por otro método que garantice obtener soluciones diversas. En las restantes iteraciones del método se realiza un proceso de contracción de la malla, que se basa en una selección de los nodos con mejor calidad entre los nodos ( $N$  nodos) existentes al final de cada iteración.

#### **2.2.6.3 Generación de nodos en dirección a los extremos locales:**

El primer tipo de exploración que se realiza en VMO se lleva a cabo en las vecindades de cada uno de los nodos de la malla inicial. Para lo cual, se buscan los vecinos más cercanos de cada nodo  $n$  a través de una función de distancia o semejanza, en dependencia del dominio de cada dimensión.

Luego, se selecciona cuál de los vecinos tiene mejor calidad (evaluación de la FO) que el nodo actual (denotándose ese mejor nodo por  $n_l$ ). Si ninguno de los vecinos es mejor, entonces este se considera un extremo local y no se generan nodos a partir de él en este paso. En otro caso, se genera un nodo ( $n^*$ ) que estará situado entre el nodo ( $n$ ) y el extremo local ( $n_l$ ).

La cercanía del nuevo nodo al actual o al extremo local depende de un factor ( $r$ ), calculado en base a los valores que alcanza la FO en cada uno de los nodos involucrados.

Luego se calculan los valores de las componentes del nuevo nodo usando la ecuación:

$$n^*(i) = f(n(i), nl(i), r) \quad (2.4)$$

Donde  $i$  representa el  $i$ -ésimo componente de cada nodo. La función  $f$  depende totalmente del dominio de cada una de las dimensiones del problema, por lo que se puede presentar como una función de selección de valores para casos discretos o como una aproximación para casos continuos. Mientras mayor sea la diferencia entre los valores de FO en los nodos involucrados, mayor será la cercanía o semejanza de  $n^*$  a  $nl$ , esto lo garantiza el factor ( $r$ ). Este paso se encarga de realizar una intensificación de la búsqueda en la vecindad de cada nodo.

#### **2.2.6.4 Generación de nodos en dirección hacia el extremo global:**

Este paso tiene como propósito realizar una exploración global hacia el nodo que mejor calidad ha tenido hasta el momento (extremo global,  $ng$ ); para lo cual, se generan nuevos nodos a partir de cada nodo de la malla inicial en dirección a este utilizando la ecuación:

$$n^*(i) = g(n(i), ng(i), r) \quad (2.5)$$

Al igual que en el paso anterior,  $g$  es una función que está totalmente relacionada con el dominio de cada dimensión y garantiza que mientras mayor sea la diferencia entre la calidad de cada nodo involucrado (determinado por  $r$ )

mayor será la cercanía del nuevo nodo al extremo global. Este paso es el encargado de acelerar la convergencia del método.

Generación de nodos a partir de los nodos más externos de la malla: este proceso de generación de nuevos nodos tiene lugar con el objetivo de explorar el espacio de búsqueda en dirección a las fronteras de cada dimensión. Para ello, se seleccionan los nodos cuyas posiciones se encuentran en los extremos de la malla inicial (nodos fronteras). El proceso de detección de este tipo de nodo ( $nf$ ) se realiza siguiendo algún criterio en dependencia del espacio solución. La generación de los nuevos nodos se obtiene a través de la ecuación:

$$n^*(i) = h(nf(i), w) \quad (2.6)$$

Donde  $w$  se conoce como desplazamiento y tiene como objetivo desplazar los nodos fronteras en dirección a los puntos más y menos externos del espacio de soluciones (depende de la definición de las fronteras del problema).

Nótese que en este paso se seleccionan tantos nodos externos como sean necesarios para completar el tamaño de la malla en el ciclo; en principio, se debe tener:

$$(N > 3 * Ni) \quad (2.7)$$

Para garantizar que se generen algunos nodos en este paso.

Como se puede apreciar, este paso explora en los entornos de los nodos que definen las fronteras de la malla en cada iteración y es utilizado para ampliar la búsqueda hacia las fronteras del espacio de búsqueda [34].

## **2.2.7 Software para Redes Bayesianas**

En la actualidad existen variados productos de software para el uso de modelos probabilísticos como lo son las redes bayesianas. En un principio resultaron muy costosos debido a que fueron concebidos bajo grandes proyectos de investigación, de los cuales se pueden nombrar a Netica, Hugin, Elvira, Weka. Netica y Hugin al ser una versión de software propietario, solo se puede utilizar una versión gratuita de prueba que no permiten utilizar todas sus bondades.

En cambio el software Elvira al ser un software de código abierto se han podido implementar extensiones que le permiten una interacción completa con el software Weka.

Para la realización de esta investigación se hizo necesaria la utilización del software de código abierto Elvira al cual se le agregaron clases adicionales para probar el algoritmo de optimización de mallas variables que se presenta como solución al problema de entrenamiento estructural de redes bayesianas.

## **2.2.8 Software Elvira**

Las redes bayesianas surgieron en la década de los 80 como modelo probabilístico para el razonamiento con incertidumbre en inteligencia artificial. En pocos años experimentaron una notable expansión: se formaron grupos especializados en las universidades más importantes (UCLA, Stanford, MIT, Carnegie-Mellon...) y en las grandes compañías (IBM, Microsoft, Digital, etc.).

También en España surgieron grupos de investigadores en varias universidades, los cuales decidieron unirse para solicitar un Proyecto Coordinado de I+D financiado por la CICYT, que se ha desarrollado entre los años 1997 y 2000. En él participaron 25 profesores de 8 universidades españolas, agrupados en cuatro sub-proyectos:

- Granada
- Almería
- País Vasco
- UNED.

La concesión de una Beca para la incorporación de tecnólogos a proyectos de investigación de la Comunidad de Madrid supuso una gran ayuda para este proyecto.

El principal objetivo del proyecto era la construcción de un entorno que sirviera, por un lado, para la investigación de nuevos métodos y algoritmos de razonamiento probabilístico y, por otro, para la implementación de sistemas expertos bayesianos. El programa resultante se llamó Elvira, tomando el antiguo nombre de la ciudad de Granada, a cuya Universidad están vinculados en mayor o menor medida varios de los investigadores del proyecto. Por la misma razón, el proyecto de la CICYT asociado fue denominado entre sus participantes como Proyecto Elvira.

En marzo de 2001, un grupo formado más o menos por los mismos investigadores solicitó un nuevo Proyecto Coordinado, titulado Elvira II: Aplicaciones de los Modelos Gráficos Probabilísticos, que fue concedido por el Ministerio de Ciencia y Tecnología a finales de ese mismo año. Sus objetivos principales son dos: mejorar las características del programa Elvira actual y desarrollar aplicaciones en diversos campos, como la medicina, la genética, la agricultura y el comercio inteligente (filtrado cooperativo).

El proyecto Elvira II se compone de cinco sub-proyectos:

- Granada

- Almería
- País Vasco
- UNED
- Albacete.

El programa Elvira cuenta con un formato propio para la codificación de los modelos, un lector-intérprete para los modelos codificados, una interfaz gráfica para la construcción de redes, con opciones específicas para modelos canónicos (puertas OR, AND, MAX, etc.), algoritmos exactos y aproximados (estocásticos) de razonamiento tanto para variables discretas como continuas, métodos de explicación del razonamiento, algoritmos de toma de decisiones, aprendizaje de modelos a partir de bases de datos, fusión de redes, etc.

Elvira está escrito y compilado en Java, lo cual permite que pueda funcionar en diferentes plataformas y sistemas operativos (linux, MS-DOS/Windows, Solaris, etc.).

### **2.2.9 Técnicas estadísticas utilizadas para el proceso de validación de los resultados**

No existe un procedimiento establecido y aceptado para comparar algoritmos sobre múltiples conjuntos de datos. Un motivo importante es el comportamiento no determinístico de éstos, por lo que la diferencia detectada entre los resultados de dos algoritmos podría deberse a factores aleatorios, y no a una mejora real [35] [36].

Para intentar determinar si las diferencias encontradas entre dos algoritmos son significativas, los investigadores pueden aplicar distintas técnicas estadísticas. A lo largo de este trabajo de investigación se aplican normalmente

las pruebas no paramétricas [37], debido a que los resultados obtenidos por las meta-heurísticas no cumplen las condiciones requeridas para poder usar de forma correcta comparaciones paramétricas [38].

A continuación se resume el proceso seguido para identificar si hay uno o varios algoritmos significativamente mejores que otros; en todos los casos se trabaja con un margen de error del 5% (valor de significación de 0.05); en caso de utilizar otro será especificado.

1. Aplicar la prueba de Iman-Davenport [39] para detectar diferencias entre un conjunto de algoritmos.
2. Si no se detectan diferencias, se puede concluir que los algoritmos involucrados obtienen resultados que no difieren significativamente unos de otros.
3. Se utiliza una prueba de Holm [40] con el conjunto de algoritmos parecidos, para corroborar los resultados de la prueba anterior. Esto se realiza porque en [38] se presenta Holm como una prueba más potente que Bonferroni-Dunn, pero más costosa
4. En caso de que sea necesario establecer una comparación entre dos algoritmos, se realiza una prueba de Wilcoxon [41].

A continuación se describen las pruebas aquí utilizadas, conjuntamente con la forma de interpretar los resultados de cada una de las tablas y las figuras involucradas. Para más información remitirse a [38] donde se presenta con total profundidad el marco experimental utilizado en este trabajo.

### **2.2.9.1 Test de Friedman:**

Se trata de un análisis de varianza de segunda vía para variables apareadas u de carácter no paramétrico. Calcula el orden de los resultados observados por

el algoritmo ( $r_j$  para el algoritmo  $j$  con  $k$  algoritmos) para cada función, asignando al mejor de ellos el orden 1, y al peor el orden  $k$ . Bajo la hipótesis nula, que se forma a partir de suponer que los resultados de los algoritmos son equivalentes y, por tanto, sus rankings son similares, el estadístico de Friedman

$$x_F^2 = \frac{12N}{k \cdot (k+1)} \left[ \sum j R_j^2 - \frac{k \cdot (k+1)^2}{4} \right] \quad (2.8)$$

Se distribuye aproximadamente acorde a  $x^2$  con  $k-1$  grados de libertad, siendo  $\frac{1}{N} \sum_i r_i^j$ , y  $N$  el número de casos. Los valores críticos del estadístico de Friedman coinciden exactamente con los establecidos en la distribución  $x^2$  cuando  $N > 10$  y  $k > 5$ . Aunque se pudo haber aplicado el test de Friedman tal y como fue definido, se decidió aplicar finalmente una variante suya, conocida como el test de Iman-Davenport [39]. El motivo es que sus autores demostraron que la función de Friedman  $x_F^2$  era excesivamente conservativa y definieron una variante con un mejor comportamiento, indicada en la ecuación siguiente:

$$F_F = \frac{(N-1)x_F^2}{N \cdot (k-1) - x_F^2} \quad (2.9)$$

Bajo las mismas condiciones de equivalencia el valor de Iman-Davenport se distribuye acorde a una distribución  $F$  con  $k-1$  y  $(k-1)(N-1)$  grados de libertad. Por tanto si el valor del test es mayor que el valor correspondiente de la distribución  $F$  (Tabla B.4 en [42]), entonces esto quiere decir que existe al menos una diferencia estadística relevante entre los algoritmos considerados.

El test de Iman-Davenport, como el de Friedman, sólo indica que existe alguna diferencia significativa, no cuál o cuáles algoritmos ofrecen diferencias estadísticamente relevantes [43] [44].

Para eso, existe una serie de test de comparaciones múltiples (post-hoc) que pueden ser aplicados en ese caso para identificar entre qué algoritmos existe una diferencia relevante.

Dos de estos tests son los siguientes.

### 2.2.9.2 Test de Holm:

Un test que prueba secuencialmente las hipótesis ordenadas según su significación. Los valores de p se ordenan por  $p_1, p_2, \dots$ , de tal forma que  $p_1 \leq p_2 \leq \dots \leq p_{k-1}$ . El método de Holm compara cada  $p_i$  con  $\alpha/(k-i)$  comenzando desde el valor de p más significativo. Si  $p_1$  es menor que  $\alpha/(k-1)$ , la correspondiente hipótesis se rechaza y nos permite comparar  $p_2$  con  $\alpha/(k-2)$ . Si la segunda hipótesis se rechaza, se continúa el proceso de comparación. En cuanto una determinada hipótesis no puede ser rechazada, todas las restantes se mantienen como aceptadas. El estadístico para comparar el algoritmo  $i$ -ésimo con el  $j$ -ésimo es:

$$z = \frac{(R_i - R_j)}{\sqrt{\frac{k \cdot (k+1)}{6 \cdot N}}} \quad (2.10)$$

El valor  $z$  se utiliza para encontrar la probabilidad correspondiente a partir de la tabla de la distribución normal, la cual es comparada con el correspondiente valor de  $\alpha$ . En [38] se prueba como el test de Holm es más potente que Bonferroni-Dunn y no hace ninguna suposición adicional sobre las hipótesis chequeadas en [40].

Se trata de una alternativa no paramétrica al t-test de muestras apareadas. Su funcionamiento se basa en calcular las diferencias entre los resultados de dos algoritmos y calcular un ranking utilizando dicho valor, a través de todas las comparaciones. Nótese que en este caso, el ranking  $d$  va desde 1 hasta  $N$ , en vez de hasta  $k$ , como era el caso de los tres tests anteriores. Tras sumar los rankings diferenciándolos entre si son negativos o positivos, se obtienen dos

valores  $R^+$  y  $R^-$ . Si el menor de ellos es menor o igual al valor de la distribución T de Wilcoxon para N grados de libertad (Tabla B.12 en [42]), se rechaza la hipótesis nula, y el algoritmo asociado al mayor de los valores es el mejor. Seguidamente se presenta la estructura de las tablas involucradas en este proceso estadístico, para lo cual se asume que se están comparando 5 algoritmos (A1, A2, A3, A4, A5) [41].

Tabla 3. Representación del test iman-davenport

Método	Valor del Test	Valor de la distribución	Valor p	Hipótesis
Iman-Davenport	25,000	2,561	5,12E-9	R

La Tabla 3 muestra los resultados del test de Iman-Davenport, donde la segunda columna representa los valores del test y la tercera columna el valor de la distribución (FF para Iman-Davenport). Este valor es extraído de una tabla distribución, en dependencia de los grados de libertad. Si el valor del test es mayor que el valor de la distribución la hipótesis de igualdad es rechazada (R) (columna 5) y por tanto fueron detectadas diferencias significativas con el grupo de control, en caso contrario se acepta (A). La columna 4 muestra el valor p computado por este test, con el cual también se puede saber si se acepta o rechaza la hipótesis (si éste es menor o igual que el margen de error permitido, la hipótesis se rechaza caso contrario se acepta).

Tabla 4. Ejemplo de un test de holm para  $\alpha_1$  como algoritmo de control

Algoritmo	z	Valor p	$\alpha_i$	Hipótesis
A3	2,325	0,0200	0,0250	R
A2	1,923	0,0544	0,0500	A

En este caso la Tabla 4 contiene los resultados del test del Holm para los algoritmos que según Bonferroni-Dunn no presentan diferencia (por debajo del valor crítico). Donde el algoritmo seleccionado como de control es el que menor

orden medio obtuvo en la Figura 1. La columna identificada como z es el valor obtenido por el test. La siguiente es el valor p asociado al z correspondiente a partir de una distribución normal, y a continuación el valor de  $\alpha/i$  que se ha de comparar. Finalmente, la última columna muestra si se rechaza (R) la hipótesis asociada a la fila o se acepta (A).

De dicha tabla se puede observar que se identifica el algoritmo A3 como estadísticamente peor que el algoritmo de control A1 y en el caso del algoritmo A2 los resultados son similares al control.

El aceptar o rechazar la hipótesis se puede identificar observando la relación entre el valor p y el valor de  $\alpha/i$  para un algoritmo determinado. Si la relación es menor o igual la hipótesis es rechazada a favor del algoritmo de control, es caso contrario se acepta.

Por último la Tabla 5 muestra los resultados del test de Wilcoxon, donde las columnas R+ y R- representan los valores acumulados de rangos positivos y negativos detectados por el test en el proceso de comparación. El valor del test es presentado por el la columna Valor p y por último la columna Hipótesis muestra si se rechaza (R) o se acepta (A) la hipótesis de igualdad para un valor de significación determinado. Si el test es capaz de detectar diferencias significativas, este favorece al primer algoritmo si  $R^- < R^+$ , en caso contrario las evidencias apuntan a que el segundo algoritmo es estadísticamente mejor que el primero. Para este ejemplo, el test encuentra diferencias significativas a favor del algoritmo A1.

Tabla 5. Ejemplo del test de wilcoxon

Algoritmo	R <sup>+</sup>	R <sup>-</sup>	Valor p	Hipótesis
A1 vs A2	72,00	19,00	0,042	R

## **2.3 Marco Referencial**

En la actualidad existen varias propuestas de algoritmos de entrenamiento estructural de redes bayesianas, el presente estudio se llevará a cabo para definir otra forma de aprendizaje estructural basado en la búsqueda de estructuras a través de modelos meta-heurísticos evolutivos.

Dentro de este campo podemos encontrar múltiples opciones de algoritmos que se dedican a obtener la estructura que mejor resultados arroje, bajo este concepto a continuación citaremos los algoritmos que serán comparados con nuestra propuesta VMO y que han sido probados en estudios anteriores y presentados originalmente en [2].

### **2.3.1 Algoritmo ByNet**

Según [2] este algoritmo fundamenta su procesamiento en la obtención de árboles de decisión basándose en la técnica CHAID. Esta técnica fue concebida como un método de segmentación, se llegó a consolidar como una técnica que permite subdividir una población a partir una variable clase, pero esta técnica también se puede utilizar para:

- Realizar una selección de atributos en caso de que así lo requiera.
- Entender la importancia de los atributos.
- Comprender la iteración de los atributos entre sí.
- Tomar como referencia numerosas tablas de contingencia, generadas con variables que no difieren significativamente.

Para el correcto funcionamiento de este algoritmo son fundamentales los siguientes parámetros:

- **Número máximo de significancia con respecto al Chi-Cuadrado:** es el número máximo de probabilidad en referencia a la tabla del estadístico Chi-Cuadrado, este número es considerado como una iteración.
- **La mínima cantidad de instancias para la población:** este parámetro determina la cantidad mínima de casos que una población puede contener para su posible subdivisión.
- **Número máximo en profundidad:** es la cantidad máxima de arcos que puede tener la estructura generada.

### 2.3.2 Algoritmo BayesCHAID

Según [2] el algoritmo parte de ideas propias de la técnica de segmentación de CHAID con adaptaciones para la generación de topologías más complejas que se ajustan a Redes Bayesianas.

La función principal del algoritmo es realizar una búsqueda completa de interacciones realizadas entre las variables, tanto a lo ancho como a profundidad en el árbol de posibles interacciones.

Los parámetros a utilizar en este algoritmo son los mismos que en el algoritmo anterior, aumentándole la cantidad de padres que podrían llegar a tener cada nodo de la red por generar. En estudios recientes se pudo llegar al valor máximo de 15 cuando se tienen arquitecturas de 32 bits.

Para el buen desempeño de este método se necesitan de dos estructuras fundamentales:

- Una lista completa de todas las subpoblaciones que se generaron por interacciones de Chi-Cuadrado, estas subpoblaciones deben contener toda la información relacionada con los nodos padres.
- Una matriz en la cual se detalla la estructura de la red formada.

### **2.3.3 Algoritmo BayesPSO**

Según [2] PSO es una meta-heurística de optimización estocástica basada en una población. Dentro del funcionamiento de este algoritmo se puede definir a un enjambre como una colección estructurada de agentes. La principal razón de ser de este algoritmo no está netamente en los individuos, más bien la podemos encontrar en las acciones que realizan todo el colectivo.

Es por esto que se trata de representar cada organismo como un punto en el espacio de búsqueda el cual forma su propio ajuste basado en su experiencia y la del resto de la bandada. Dentro del proceso de modelación se simboliza cada partícula como la estructura de una red bayesiana. Para este proceso se debe de tomar en consideración la métrica que se utiliza para la medición de la calidad de la red.

## **CAPÍTULO III**

# **METODOLOGÍA DE LA INVESTIGACIÓN**

### 3.1. Materiales

#### 3.1.1. Equipos Y Materiales

Los recursos materiales necesarios para la investigación son los siguientes:

##### 3.1.1.1. Hardware

Tabla 6. Materiales hardware

CANTIDAD	MATERIAL	DESCRIPCIÓN
1	COMPUTADOR	Utilizado en la investigación, desarrollo y pruebas. - HP Pavilion dm4-2055la <b>Características:</b> - Intel Core™ i5-2450M - 6 Gb. RAM o superior - 500 Gb Disco Duro - Cd Writer
1	IMPRESORA	CANNON MP230 SERIES

##### 3.1.1.2. Software.

Tabla 7. Materiales software

CANTIDAD	MATERIAL	DESCRIPCIÓN
1	SISTEMA OPERATIVO	Microsoft Windows: versión Windows 7
1	PROCESADOR DE TEXTO.	Microsoft Office 2010 Word Power Point

### 3.1.1.3. Suministros

Tabla 8. Suministros

<b>CANTIDAD</b>	<b>MATERIAL</b>	<b>DESCRIPCIÓN</b>
6 meses	INTERNET	Plan Internet Banda Ancha De Telconet.
	MATERIALES DE OFICINA	- 1 Caja Lápices y 1 Caja Lapiceros - 1 Resma Hojas A4 - 3 CDSRw - 2DVD R - 3 Marcadores y 3 Resaltadores - 1 Borradores - 1 Reglas - 1 Perforadora - 1 Grapadora y 1 caja de grapas

### 3.1.1.4. Personal

Tabla 9. Personal

<b>PERSONAL</b>	<b>DESCRIPCIÓN</b>
Desarrollador:	Luis Enrique Moreira Zamora
Director de Tesis:	Phd. Amilkar Yudier Puris Cáceres
Asesor:	Ing. Msc. Byron Wladimir Oviedo Bayas

### 3.1.1.5. Presupuesto

Tabla 10. Presupuesto

<b>DETALLE</b>	<b>VALOR</b>
Suministros de oficina:	\$20
Internet:	\$80
Impresiones:	\$70
Libros:	\$50
Movilización:	\$30
Imprevistos:	\$50
Total	\$300

## 3.2. Métodos y Técnicas Utilizados en la Investigación

### Analítico

Este analítico nos va a permitir interpretar la situación desde el punto de vista analítico-crítico, y así poder tener una mejor comprensión del objeto del estudio y lograr el fin propuesto.

### 3.2.1. Tipo de investigación

#### Exploratoria

Esta investigación permitirá introducir nuestra propuesta dentro del problema de optimización discreto debido a que el algoritmo no había sido estudiado en este campo. Encontrando los procedimientos adecuados para obtener una mejora en comparación con otros algoritmos.

### **3.2.2. Diseño de Investigación**

#### **DISEÑO EXPERIMENTAL**

##### **Cuasi-Experimental**

En este proyecto de investigación, se aplicará este método para evaluar el impacto del algoritmo basado en mallas variables.

Se considera apropiado utilizar el método cuasi-experimental de un solo grupo pre-prueba y post-prueba, con una medición antes y después de implementar la aplicación, para comprobación de la hipótesis.

#### **DISEÑO CUASI-EXPERIMENTAL DE UN SOLO GRUPO PRE-PRUEBA Y POST-PRUEBA**

Este estudio establece una medición previa a la intervención y otra posterior.  
Esquema del diseño  $O_1 X O_2$ .

Dónde:

$X$  = Algoritmo basado en mallas variables para obtener un modelo óptimo de Redes Bayesianas

$O_1$  = Medición antes del experimento

$O_2$  = Medición después del experimento

El diseño consiste entonces, en tomar un algoritmo que haya sido probado en ambientes con grandes volúmenes de datos discretos, y realizar un análisis del resultado de una red bayesiana con el algoritmo propuesto.

### 3.3. Población y Muestra

En este apartado se detalla las bases del conocimiento que han sido utilizadas para realizar las pruebas de todas las variantes del algoritmo, estas bases del conocimiento se encuentran en la UCIML [5], un repositorio muy conocido a nivel mundial y que aloja gran cantidad de bases del conocimiento, de las cuales han sido seleccionadas 14 las cuales se detallan en la tabla 11.

Tabla 11. Bases del conocimiento utilizadas

<b>BASE DEL CONOCIMIENTO</b>	<b>RASGOS DISCRETOS</b>	<b>RASGOS CONTINUOS</b>	<b>CLASES</b>	<b>CASOS</b>	<b>DISTRIBUCIÓN POR CLASE</b>
mammographic	4	1	2	961	516 545
Lung-cancer	56	0	3	32	9 13 10
hepatitis	13	6	2	155	32 123
e colic	0	7	8	336	143 77 52 35 20 5 2 2
breast-cancer-w	10	0	2	683	444 239
contac-lenses	4	0	3	24	4 5 15
hayes-roth-m	4	0	3	132	51 51 30
Monk	1	6	0	2	229 186
vote	16	0	2	300	116 184
Balance-scale	4	0	3	625	288 49 288
tic-tac-toe	9	0	2	958	626 332
Iris	0	4	3	150	50 50
Labor	8	8	2	57	20 37
Soybean	35	0	19	683	-

## **CAPÍTULO IV**

### **RESULTADOS Y DISCUSIÓN**

En el presente capítulo se detalla la manera en que se aplicó la Meta heurística VMO al problema de entrenamiento estructural de redes bayesianas, se realiza un ajuste de parámetros del modelo y posteriormente un estudio comparativo entre los resultados obtenidos por VMO con los de un conjunto de clasificadores bayesianos de los más utilizados en el estado del arte.

## 4.1 Elementos de la modelación

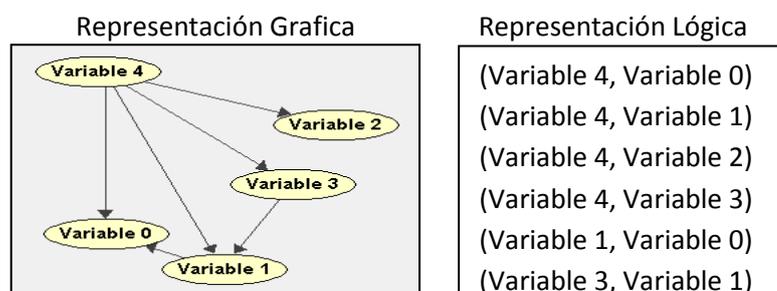
En esta sección describiremos con más detalles la forma en que se ajustaron algunos de los elementos de la meta-heurística VMO para poder ser aplicada al problema de entrenamiento estructural de redes bayesianas.

### 4.1.1 Definición del nodo

En el desarrollo de la implementación se tomó el concepto original de formar una malla de nodos que representen las posibles soluciones. Dentro del contexto de las Redes Bayesianas se logró definir a un nodo como una topología de red, que a su vez representa una posible solución al problema de optimización, en este caso utilizamos la clase Bnet.java que viene incorporado en el Elvira, la cual nos permite representar una red bayesiana tanto de forma gráfica como lógica, debido a que en algunos pasos del algoritmo se trabaja con los arcos de la red para realizar las operaciones de nodos.

Ilustración 1. Definición de un nodo

#### NODO



### 4.1.2 Cálculo de la distancia entre nodos

La distancia entre dos nodos representa cuán parecidas son dos redes a partir de sus estructuras. Para esto se realiza un simple conteo de la cantidad de arcos iguales que presentan ambas redes. Para almacenar las distancias entre todos los nodos de la malla se utilizó una matriz cuadrada de distancias donde la celda (i, j) representa la distancia entre el nodo i y el nodo j y se cumple que  $d(N_i, N_j) = d(N_j, N_i)$ . A continuación se describe el algoritmo para el cálculo de las distancias entre dos nodos:

#### Pseudocódigo 1 Cálculo De Distancia Entre

##### Calcular Distancia (N1 , N2)

1. **inicializar**  $distancia(N1, N2) \leftarrow 0$
2. **Para cada arco**  $a_{1,i} \in N1$  **hacer**
  - 2.1 **Si**  $a_{1,i} \in N2$  **entonces**
    - 2.1.1 **Incrementar**  $distancia(N1, N2)$
3. **Retornar**  $distancia(N1, N2)$

Nodos

De manera general este procedimiento para calcular distancias es invocado para obtener las distancias entre todos los pares de nodos de la malla.

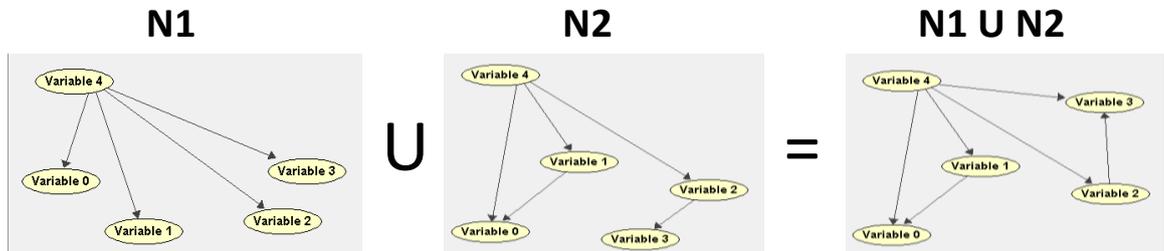
### 4.1.3 Generación de nuevas redes a partir de un par

Este elemento es la base fundamental de la fase de expansión de la malla ya que define la forma en que se van a generar las nuevas soluciones a partir de dos existentes. Para el caso de estudio de entrenamiento de redes bayesianas se definieron dos formas distintas utilizando la teoría de conjuntos, teniendo en cuenta que una red no es más que un conjunto de arcos. A continuación se detallan:

- a) **Unión entre conjuntos:** con esta operación se obtiene una nueva red la cual contiene los arcos de las redes que se unieron. De manera que se

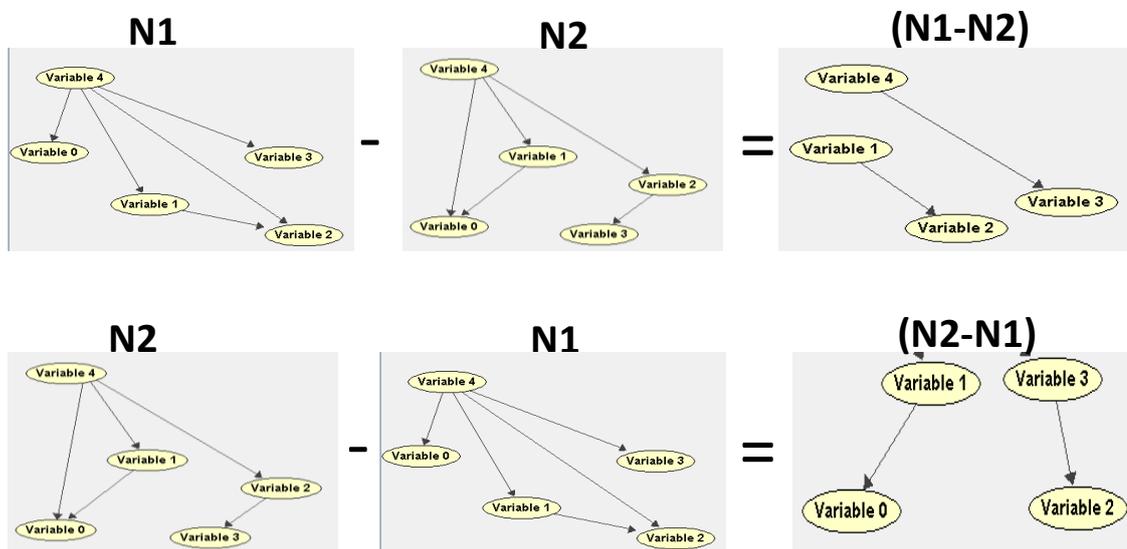
obtiene una red más general con las características de ambas. La ilustración 1 muestra un ejemplo de esta operación.

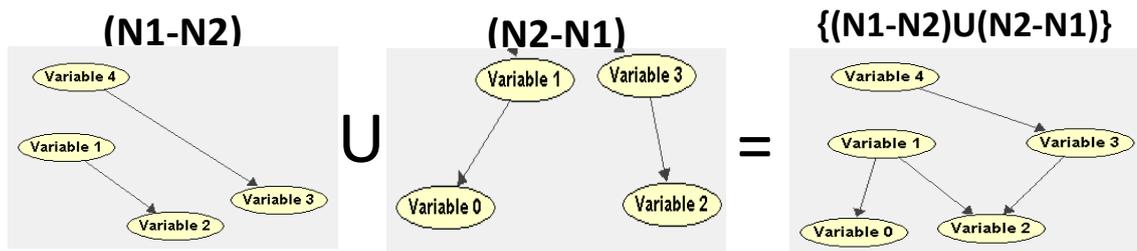
Ilustración 2. Ejemplo de la unión de conjuntos



b) **Unión de las diferencias:** Esta operación define una nueva red en base a los arcos en que difieren ambas redes. La ilustración 2 muestra un ejemplo de esta operación.

Ilustración 3. Ejemplo de la unión de diferencias





#### 4.1.4 Selección de las mejores redes

Este elemento consiste en seleccionar las redes de la malla inicial que serán utilizadas en la próxima iteración del algoritmo. Para este caso de estudio se utilizaron dos variantes:

- a) **Elitista:** consiste en ordenar de mayor a menor los nodos basándonos en la calidad de clasificación de cada red (esta información se describirá más adelante) y seleccionar los M mejores nodos.
- b) **Representativa:** se introduce un parámetro  $\xi$  que representa una distancia máxima a tener en cuenta entre los nodos que serán seleccionados. El proceso se define como sigue:

#### Pseudocódigo 2. Selección representativa

1. Para cada  $n_i, n_j \in M^{temp}$  (calidad ( $n_i$ ) mejor que calidad ( $n_j$ )) hacer
  - 1.1 Si  $\text{distancia}(n_i, n_j) > \xi$  entonces
    - 1.1.1 Eliminar  $n_j$  de  $M^{temp}$
2. Retornar  $M^{temp}$

El parámetro  $\xi$  representa el umbral que define la separación permitida entre nodos y su valor decrece a medida que aumenta la ejecución del algoritmo. En la ecuación 4.1 se define como el comportamiento de este parámetro:

$$\xi = \begin{cases} 0.8 * A, & \text{si } k < \frac{1}{4} * I \\ 0.7 * A, & \text{si } k < \frac{2}{4} * I \\ 0.6 * A, & \text{si } k < \frac{3}{4} * I \\ 0.5 * A, & \text{otros casos} \end{cases} \quad (4.1)$$

Donde  $A$  es la cantidad de atributos con la cual se conforma la base del conocimiento utilizada,  $k$  es el número de iteraciones actuales e  $I$  es el número total de iteraciones.

## 4.2 Integración del algoritmo VMO en el software Elvira

El modelo VMO es una meta-heurística que, al igual que los algoritmos genéticos o cualquier meta-heurística poblacional, presenta estrategias de exploración del espacio de búsqueda. Como el aprendizaje estructural es encontrar en un espacio de diferentes topologías la que más se adecua a los datos, por eso es que utilizamos esta estrategia. Para la aplicación de VMO al aprendizaje de redes bayesianas, utilizamos algunas de las clases que ya tiene implementado el Elvira de manera que nos resultaba más fácil reutilizar el código que este presenta.

**Paso 1 (Generación malla inicial):** se genera una población inicial de  $M$  nodos (redes) generadas de manera aleatoria mediante la clase principal VMO.java esta clase tiene el método `structuralLearning()` el cual nos permite generar aleatoriamente las redes iniciales, a cada una de las redes obtenidas se les calcula la calidad de clasificación utilizando el método `CalculatePrecision(Bnet Net, int k, int type)` de la clase principal VMO.java el cual recibe como parámetros una red bayesiana, un indicador  $k$  que sirve para dividir el conjunto de datos para hacer el proceso de validación cruzada y el tipo de generación que me permite saber en qué paso de la actual iteración se generó la red, este método anteriormente citado se conecta con el método `kFoldSumCrossValidation(int k)` propio del software Elvira el cual nos facilita el

cálculo de la precisión con el método de validación cruzada. Este cálculo es almacenado en una variable clase que nos facilita en pasos más adelante saber la precisión de la red sin tener que volver a calcularla.

**Paso 2 (Generación por extremo local - LocalExtreme(TamNeigh)):** para este paso se calcula una matriz de distancias que obtiene valores dependiendo de la estructuras de las redes comparadas, es decir comparamos los arcos que tiene el nodo A con los arcos que tiene el nodo B, este proceso lo realiza el método HammingMatrix() ubicado en la clase VMO\_MESH.java, básicamente implementamos una distancia entre conjuntos (cantidad de arcos iguales) luego se selecciona entre las redes vecinas a una, la de mejor calidad y se obtiene una nueva red entre esta y la actual a través de los métodos BestNetUnion(Vector vector, int k,int type) y DiferenciaUnion(Vector vector, int k,int type) que reciben como parámetro un vector de las mejores redes, un indicador k y el tipo de generación, para la unión entre conjuntos.

**Paso 3 (Generación por extremo global - GlobalExtreme()):** se selecciona la red de mejor calidad entre todas las de la malla y se utiliza esta para obtener nuevas redes a través de los operadores de exploración con las redes restantes.

**Paso 4 (Generación por extremo Frontera - BorderNode()):** Se Seleccionan las redes que más y menos se parecen a las demás de la malla, luego a cada una de estas redes se le cambia el sentido de un arco seleccionado de manera aleatoria.

**Paso 5 (Seleccionar la malla inicial para la siguiente iteración - setNetSolVMO()):** para el proceso de contracción de la malla se utilizan los operadores de contracción citados en el anterior apartado.

Este proceso se repite hasta que se cumple con una cantidad definida de iteraciones.

### 4.3 Aplicación del algoritmo VMO al aprendizaje estructural de Redes Bayesianas

A continuación se presenta una explicación detallada de cómo se aplicaron cada uno de los pasos del algoritmo VMO al aprendizaje de redes bayesianas. En cada paso se agrega el pseudocódigo que lo representa donde  $M$  detalla la malla inicial de cada interacción y  $M^{temp}$  una malla auxiliar donde se almacenan todos los nodos obtenidos en el proceso de expansión.

#### 4.3.1 Generación de la malla inicial en cada iteración

La malla inicial es generada aleatoriamente partiendo desde la variable clase y asignando hijos a dicha variable que luego se convierten en padres de otras variables, todo este proceso se lleva a cabo con la generación de números aleatorios que nos indica si son padres o hijos. Su tamaño varía según el requerimiento y el análisis de parámetros que realizamos más adelante, para las iteraciones subsiguientes la malla inicial resultante será la selección de las mejores redes mediante los métodos citados en el apartado 4.1.1.4.

##### Pseudocódigo 3. Generación de la malla

<p><b>Generar</b> <math>M</math> de manera aleatoria si es la primera iteración del algoritmo <b>Inicializar</b> <math>M^{temp} \leftarrow M</math> <b>Identificar</b> la mejor solución <math>n_g \in M</math></p>
---

inicial

#### 4.3.2 Generación de nodos en dirección a los extremos locales

En este paso se toman las redes iniciales y se genera una matriz de distancias descrita en 4.1.1.2 para luego obtener las tres redes más parecidas, este proceso se hace para cada nodo y luego si la mejor de las vecinas es mejor que ella se generara una nueva red utilizando el proceso descrito en 4.1.1.3.

#### Pseudocódigo 4. Generación de nodos por extremo

local

<p><b>Fase 1:</b> Para cada <math>n_i \in M</math></p> <p><b>1.1 Obtener</b> el vecino de mejor calidad <math>n_i^*</math> (vecindad de tamaño <math>k</math>)</p> <p><b>1.2 Si</b> <math>n_i^*</math> es óptimo local <b>Entonces</b></p> <p>    <b>1.2.1</b> <math>n_i^{fase1} \leftarrow \text{combinar}(n_i, n_i^*)</math> (algoritmo 4.1.3)</p> <p><b>1.3</b> <math>M^{temp} \leftarrow M^{temp} \cup n_i^{fase1}</math></p>
---

### 4.3.3 Generación de nodos en dirección a los extremos globales

Se selecciona el nodo con mejor calidad y mediante el proceso descrito en 4.1.1.3 se generan nuevas redes con los nodos de la malla inicial.

#### Pseudocódigo 5. Generación por extremo

global

<p><b>Fase 2:</b> Para cada <math>n_i \in M</math></p> <p><b>2.1</b> <math>n_i^{fase2} \leftarrow \text{combinar}(n_i, n_g)</math> (algoritmo 4.1.3)</p> <p><b>2.2</b> <math>M^{temp} \leftarrow M^{temp} \cup n_i^{fase2}</math></p>
---

### 4.3.4 Generación de nodos en dirección de frontera

El término frontera es utilizado en este trabajo para identificar los nodos que más y menos se asemejan al resto de los nodos. Para esto, se utiliza la matriz de distancia y se seleccionan los nodos de mayor y menor distancia promedio. Luego se le aplica una pequeña alteración a la dirección de uno de los arcos seleccionado de manera aleatoria de cada uno de los nodos.

## Pseudocódigo 6. Generación por extremos frontera

**Fase 3:** Expandir a través de la frontera

**3.1 Seleccionar**  $n_s \in M$  (mayor distancia promedio)

**3.2**  $n_s^{fase3} \leftarrow \mathbf{transformar}(n_s)$

**3.3**  $M^{temp} \leftarrow M^{temp} \cup n_s^{fase3}$

**3.1 Seleccionar**  $n_f \in M$  (menor distancia promedio)

**3.2**  $n_f^{fase3} \leftarrow \mathbf{transformar}(n_f)$

**3.3**  $M^{temp} \leftarrow M^{temp} \cup n_f^{fase3}$

### 4.3.5 Contracción de la malla

Al término de cada iteración se necesita realizar la contracción de la malla, que corresponde a seleccionar los  $p$  nodos que conformaran la malla inicial de la próxima iteración. En el epígrafe 4.1.2.2 se describe las dos variantes de selección utilizadas en este trabajo. Seguidamente se presenta la estructura general del proceso de contracción:

## Pseudocódigo 7. Contracción de la

**Fase 4:** Contracción de la malla

**4.1 Ordenar**  $M^{temp}$  de manera creciente en cuanto a calidad

**4.2 Si** selección elitista **entonces**

**4.2.1**  $M \leftarrow$  primeros  $p$  elementos de  $M^{temp}$

**4.3 Si** selección representativa **entonces**

**4.3.1**  $M^{temp} \leftarrow$  aplicar algoritmo 4.1.4. b a  $M^{temp}$

**4.3.2 Si**  $|M^{temp}| \geq p$  **entonces**

**4.3.2.1.**  $M \leftarrow$  primeros  $p$  elementos de  $M^{temp}$

**4.3.3 Sino**  $M \leftarrow M^{temp} \cup N, N$  nodos generados aleatoriamente

malla

### 4.4 Estudio experimental de VMO y análisis de los resultados

En este apartado se presenta un estudio de la propuesta orientado a encontrar la variante de VMO que mejores resultados obtiene para luego realizar una

comparación con algunos de los clasificadores bayesianos más utilizados en la literatura. Para este estudio se utilizaron 14 bases de conocimientos tomadas del repositorio de la UCIML [5], con diferentes características (Ver especificaciones en Tabla 11 del capítulo III).

La experimentación se realizó de la siguiente manera:

1. Obtener la variante que mejor resultados obtuvo para la selección elitista teniendo en cuenta las dos formas diferentes de generación de nuevas soluciones y los diferentes valores de los parámetros (tamaño de la malla inicial  $p=(12, 24, 48)$  y tamaño de las vecindades  $k=(3, 5, 7)$  y cantidad de iteraciones 1000).
2. Realizar el mismo estudio anterior pero en este caso para la selección por representatividad.
3. Seleccionar cuál de las variantes seleccionadas en los pasos anteriores es la que mejor resultados obtuvo.
4. Comparar la variante seleccionada anteriormente con un conjunto de clasificadores presentes en el estado del arte.

Basándonos en lo antes mencionado se presentan diferentes variantes del algoritmo, por lo que en la tabla 12 se detalla la nomenclatura que identifica cada una de estas:

Tabla 12. Nomenclatura utilizada en las pruebas

<b>NOMENCLATURA</b>	<b>DESCRIPCIÓN</b>
VMO (E_U,k,p)	Implementación elitista con el operador unión.
VMO (E_D,k,p)	Implementación elitista con el operador unión de diferencias.
VMO (NE_U,k,p)	Implementación representativa con el operador unión.

VMO (NE_D,k,p)	Implementación representativa con el operador unión de diferencias.
----------------	---

Donde la variable  $k$  representa el tamaño de la vecindad y  $p$  el tamaño de la malla inicia de cada iteración.

Cada variante estudiada fue ejecutada 50 veces de manera independiente para cada base de conocimiento y los valores mostrados en las tablas de resultados representan el valor promedio. Para realizar las comparaciones se utilizó un conjunto de test estadísticos (ver capítulo II) de la siguiente manera:

1. Comparaciones múltiples (entre grupos): se aplicó el test de Friedman e Iman Davenport para detectar diferencias significativas entre los grupos, sino existen se puede concluir que los algoritmos comparados tiene comportamientos parecidos, en caso que existan diferencias se procede aplicar un test de Holm o Wilcoxon según el caso.
2. Comparaciones múltiples con una muestra de control (3 o más algoritmos): se utilizó el test de Holm para comparar una muestra de control (algoritmo con menor valor medio) con los demás algoritmos, para detectar si las diferencias entre los resultados son significativas a favor del algoritmo de control.
3. Comparaciones por pares (entre dos algoritmos): se aplicó el test de Wilcoxon solo para los casos donde se necesitaba comparar exactamente dos algoritmos.

#### 4.4.1 Análisis de la sección elitista

Este apartado tiene como finalidad el estudio de parámetros para el algoritmo VMO basándonos en el método de selección elitista (ver 4.1.2.2.a) utilizando

los diferentes operadores de generación de nuevas redes (ver 4.1.2.1). A continuación se detallan los estudios realizados.

#### 4.4.1.1 Operador de Unión

El operador de Unión se encarga de combinar las características de dos redes en una sola (ver capítulo 4.1.2.1 a)). En el Anexo 1 se muestra los resultados obtenidos utilizando este operador y a continuación se realizan los análisis estadísticos necesarios para seleccionar la mejor variante.

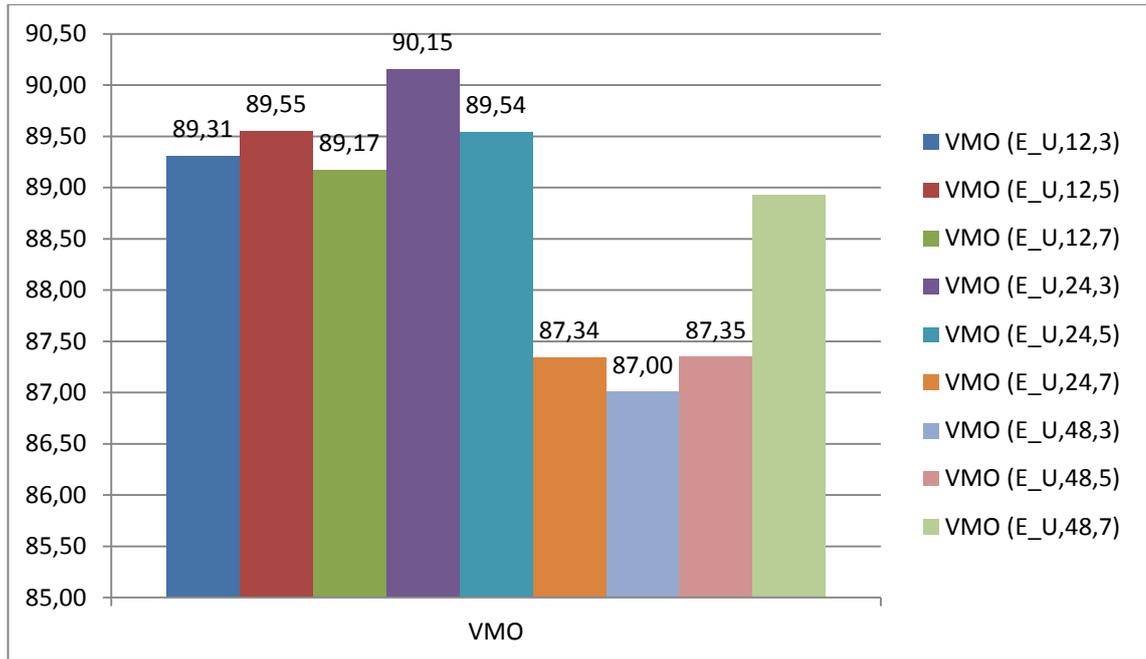
La tabla 13 presenta los resultados de aplicar el test de Friedman e Iman Davenport (ver capítulo 2), para verificar si existen diferencias significativas en los resultados obtenidos por los algoritmos computados (columnas p de la tabla 4.4). El valor de la distribución Chi Cuadrado con 8 grados de libertad para el test de Friedman se muestra en la columna “valor de la distribución”, así como el valor de la distribución de Fisher con 8 y 104 grados de libertad para el test de Iman Davenport. Para ambos casos el valor del test es mayor que el valor de la distribución por lo que se rechaza la hipótesis de igualdad de medias (columna Hipótesis) y se concluye que existen diferencias significativas entre los algoritmos comparados.

Tabla 13. Test de Friedman e Iman Davenport para VMO(E\_U,k,p)

TEST	VALOR DEL TEST	VALOR DE LA DISTRIBUCIÓN	VALOR P	HIPÓTESIS
Friedman	48.742	15,507	7.126E-8	RECHAZADA
Iman Davenport	10.017	2.032	2.92E-10	RECHAZADA

Seguidamente se aplica el test de comparaciones múltiples de Holm (ver Capítulo II) para comparar el algoritmo de mayor valor medio (VMO (E\_U,24,3), ver ilustración 3), con los demás.

Ilustración 4. Valores promedios de las pruebas para VMO(E\_U,k,p)



La tabla 14 muestra los resultados del test, donde los algoritmos han sido ordenados de forma creciente por el valor z. Los valores de la columna “Valor p” se puede encontrar utilizando la distribución normal y como su valor es menor que los presentados en la columna  $\alpha/i$ , la hipótesis de semejanza es rechazada para ambos casos (columna Hipótesis). Con este análisis se puede concluir que el algoritmo VMO (E\_U,24,3) obtiene resultados significativamente mejores que los que obtuvieron las demás variantes estudiadas en este apartado.

Tabla 14. Resultado del test de holm para VMO(E\_U,k,p)

ALGORITMO	z	Valor p	$\alpha/i$	HIPÓTESIS
VMO(E_U,48,5)	4.209	2.560E-5	0.0062	RECHAZADA
VMO(E_U,48,3)	4.174	2.981E-5	0.0071	RECHAZADA
VMO(E_U,24,7)	3.726	1.942E-4	0.0083	RECHAZADA
VMO(E_U,48,7)	2.863	0.0041	0.01	RECHAZADA
VMO(E_U,12,3)	1.276	0.201	0.0125	ACEPTADA
VMO(E_U,12,7)	1.173	0.240	0.0166	ACEPTADA
VMO(E_U,24,5)	0.310	0.756	0.025	ACEPTADA

VMO(E_U,12,5)	0.276	0.782	0.05	ACEPTADA
---------------	-------	-------	------	----------

Como pudimos observar en la tabla 14 el algoritmo de control (VMO(E\_U,24,3)) obtuvo resultados significativamente superiores sobre los cuatro primeros algoritmos citados en dicha tabla, pero en los últimos cuatro la hipótesis fue aceptada por lo que no hay diferencia significativas y se supone que los resultados son semejantes. En este caso se selecciona el algoritmo de control VMO(E\_U,24,3). Ya que es significativamente mejor que el 50% de los algoritmos comparados y alcanza mayor valor medio.

#### 4.4.1.2 Operador de Unión de las diferencias

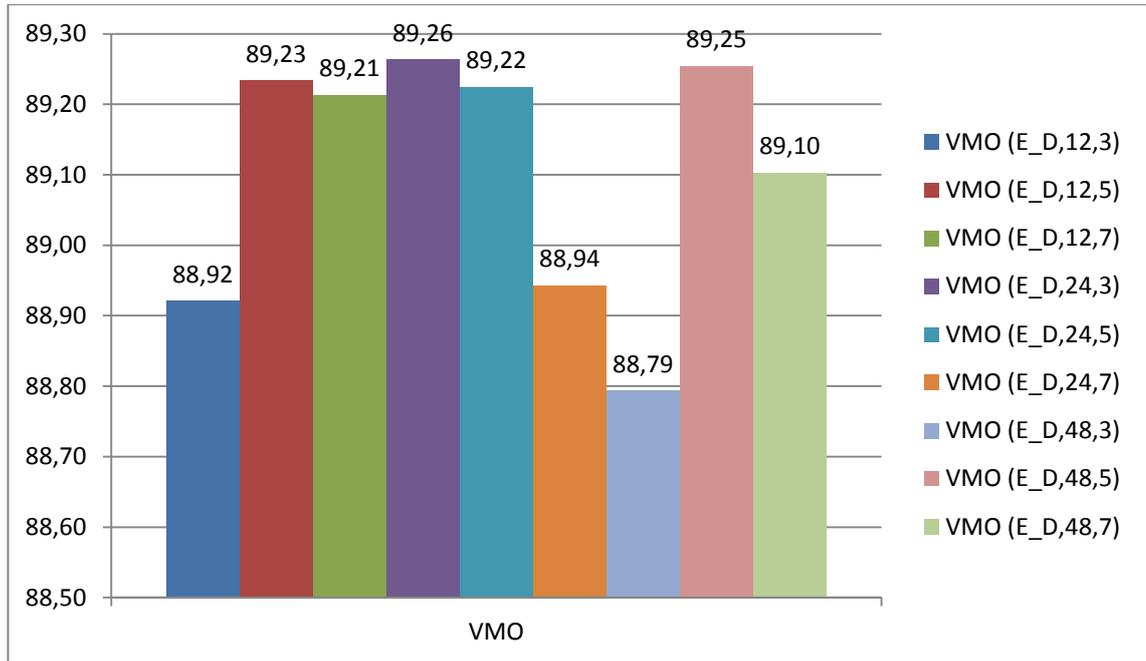
Este operador se utiliza para generar una nueva red con las características en las que difieren dos redes (ver 4.1.2.1.b)). En el Anexo 2 se muestran los resultados obtenidos con la aplicación de esta variante a las bases del conocimiento.

Tabla 15. Test de fridman e iman davenport para VMO(E\_D,k,p)

TEST	VALOR DEL TEST	VALOR DE LA DISTRIBUCIÓN	VALOR P	HIPÓTESIS
Friedman	3.466	15,507	0.901	ACEPTADA
Iman Davemport	0.415	2.032	0.909	ACEPTADA

En la tabla 15 se muestran los resultados obtenidos de los test de Fridman e Iman Davenport donde la hipótesis semejanza es aceptada por lo que no existen diferencias significativas entre los resultados alcanzados por las variantes computadas. Para este caso específicamente no se realiza test de comparaciones múltiples y se selecciona la variante VMO(E\_D,24,3) como la de mayor valor medio (ver ilustración 4).

Ilustración 5. Valores promedios de las pruebas para VMO(E\_D,k,p)



#### 4.4.2 Obtención de la mejor variante para selección elitista

Según los análisis realizado para el tipo de selección elitista, la variante que mejores resultados obtuvo tanto para el operador de unión como el de la unión de las diferencias fue VMO con  $p=24$  y  $k=3$ , en algunos casos obteniendo resultados significativamente mejores que las demás variantes y otros caso por alcanzar mayor valor medio. A continuación en la tabla 16 se presenta los resultados del test de Wilcoxon para comparar ambas variantes y seleccionar la que mejor comportamiento obtuvo para la selección elitista.

Tabla 16. Resultado del test de wilcoxon selección elitista

COMPARACIÓN	R <sup>+</sup>	R <sup>-</sup>	VALOR DEL TEST	HIPÓTESIS
VMO(E_U,24,3) vs VMO(E_D,24,3)	70	35	0.13567	ACEPTADA

Claramente se puede apreciar que el valor del test es mayor que el nivel de significancia determinado para el mismo ( $\alpha=0.05$ ) por lo que la hipótesis de igualdad es aceptada (columna hipótesis). Con estos resultados se puede

concluir que ambas variantes de la selección elitista presentan comportamientos similares, por lo que la selección de cualquiera de los dos queda totalmente justificada. En nuestro caso seleccionaremos la variante VMO(E\_U,24,3) ya que presenta mayor valor medio.

#### **4.4.3 Análisis de selección representativa**

Este apartado tiene como finalidad poner a prueba la selección por representatividad que nos da una forma diferente a la primer propuesta planteada, esta forma de selección nos permite realizar una exploración más amplia del espacio de búsqueda debido a que no se concentra solo en los mejores nodos de toda la malla sino más bien va segmentando el espacio de búsqueda.

##### **4.4.3.1 Operador de Unión**

Utilizando la variante de selección representativa junto con el operador unión, la cual nos permite obtener una red mucho más completa, con un número mayor de aristas que las de otras variantes pero explorando diferentes lugares en el espacio de búsqueda, después de aplicar las pruebas se obtuvieron los valores promedios de la media que se muestran en el anexo 3 como resultados de las pruebas realizadas.

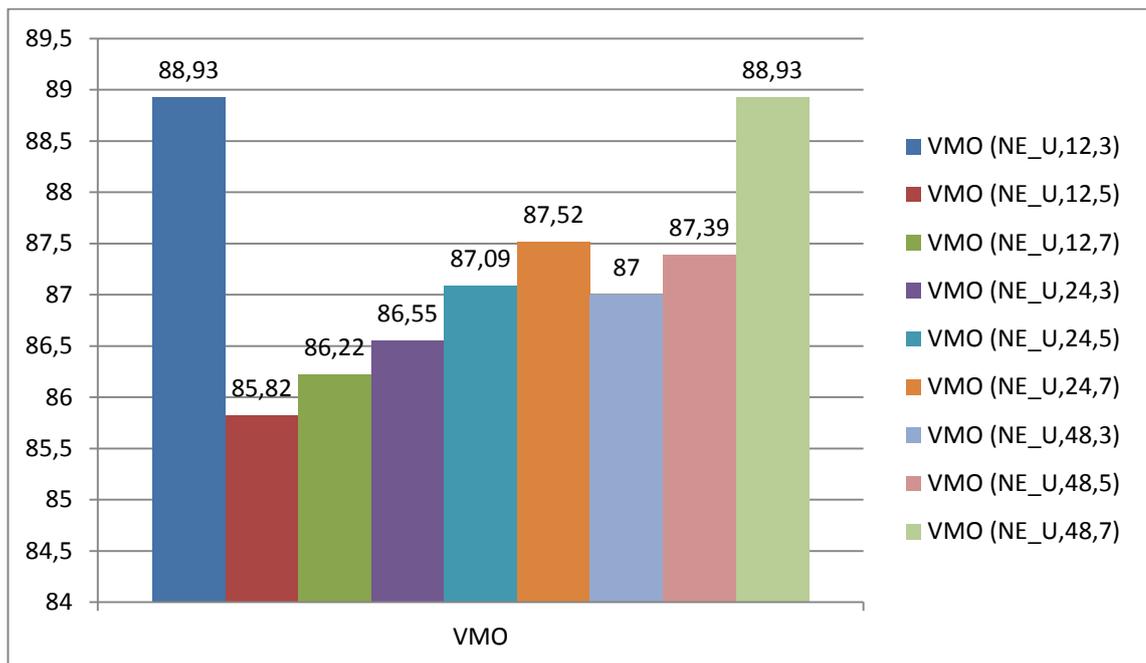
Luego de aplicarle el test de Friedman e Iman Davenport obtuvimos los resultados que se muestran en la tabla 17. Utilizando 8 grados de libertad para la distribución de Chi Cuadrado al aplicar el test de Fridman así como 8 y 104 grados de libertad en la distribución de Fisher para la aplicación del test de Iman Davenport obtuvimos que el valor resultante de los test es mayor que los valores de la distribución, por tanto se rechaza la hipótesis de igualdad de media entre las variantes aplicadas y se supone diferencias significativas en el grupo.

Tabla 17. Test de Friedman e Iman-Davenport para VMO(NE\_U,k,p)

TEST	VALOR DEL TEST	VALOR DE LA DISTRIBUCIÓN	VALOR P	HIPÓTESIS
Friedman	29.876	15,507	2.222E-4	RECHAZADA
Iman-Davenport	4.729	2.032	5.744E-5	RECHAZADA

Los resultados del test de Holm aplicado a los resultados mostrados en el anexo 3 se muestran en la tabla 4.10, con los cuales se puede concluir que el algoritmo VMO(NE\_U,48,7) es seleccionado como el algoritmo de control para comparar las demás variantes.

Ilustración 6. Valores promedios de las pruebas para VMO(NE\_U,k,p)



Como se muestra en la tabla 18, la hipótesis es rechazada en la mayoría de casos en la cual obtiene resultados significativamente mejores que los que obtuvieron las demás variantes estudiadas en este apartado, a excepción de la última variante que muestra resultados parecidos, basándonos en la ilustración 5 que nos muestra los valores medios alcanzados por los algoritmos, donde se

puede apreciar que dos variantes (VMO(NE\_U,12,3) y VMO(NE\_U,48,7)) alcanzan el mismo valor. En este caso seleccionaremos el algoritmos VMO(NE\_U,48,7) que representa el grupo VMO(N\_E,48, k), ya que el grupo presenta un valor medio de 87.77, mayor que el grupo de VMO(N\_E,12,k).

Tabla 18. Resultado del test de holm para VMO(NE\_U,k,p)

<b>ALGORITMO</b>	<b>z</b>	<b>Valor p</b>	<b><math>\alpha/i</math></b>	<b>HIPÓTESIS</b>
VMO (NE_U,12,7)	4.485	7.276E-6	0.0062	RECHAZADA
VMO (NE_U,12,5)	4.174	2.981E-5	0.0071	RECHAZADA
VMO (NE_U,24,3)	3.105	0.0019	0.0083	RECHAZADA
VMO (NE_U,24,5)	2.898	0.0037	0.01	RECHAZADA
VMO (NE_U,24,7)	2.863	0.0041	0.0125	RECHAZADA
VMO (NE_U,48,3)	2.829	0.0046	0.0166	RECHAZADA
VMO (NE_U,48,5)	2.587	0.0096	0.025	RECHAZADA
VMO (NE_U,12,3)	1.276	0.201	0.05	ACEPTADA

#### 4.4.3.2 Operador de Unión de Diferencias

Este operador toma como patrón la selección de atributos que de cierta manera nos permite reducir el tamaño de la estructura de la red, tomando como selección representativa podemos obtener un espacio de búsqueda explorado en su totalidad con una media de selección.

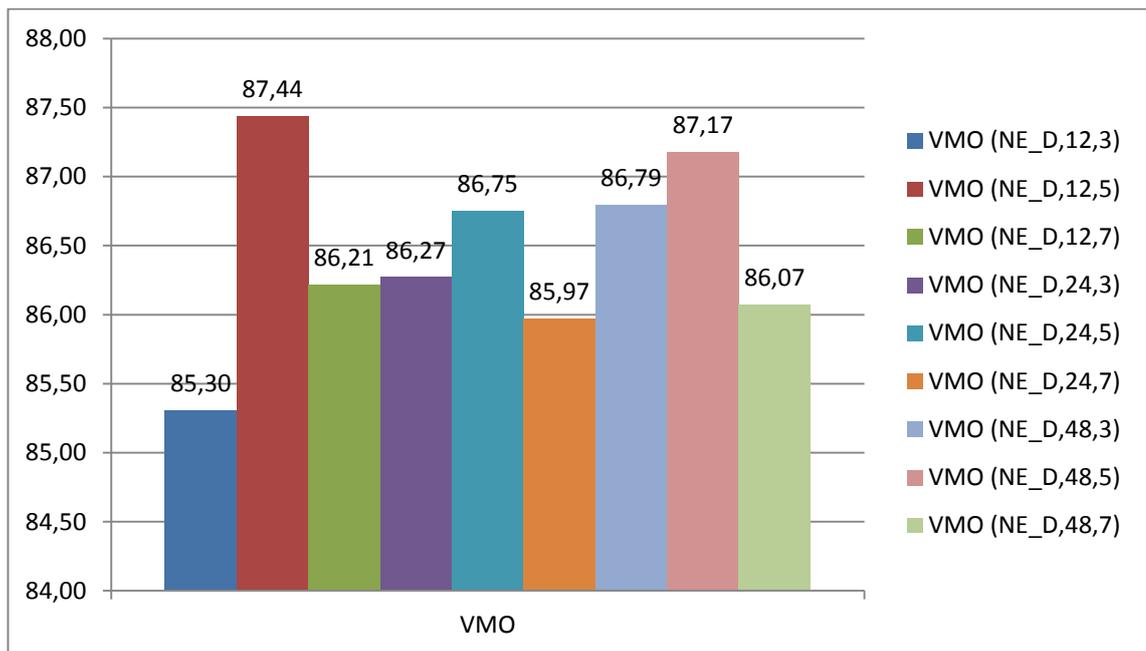
En el anexo 4 podemos visualizar los resultados del promedio de las pruebas realizadas a esta variante, se les aplicó el test de Friedman con 8 grados de libertad en la distribución Chi Cuadrado, 8 y 104 grados de libertad para la distribución de Fisher.

Tabla 19. Test de fridman e iman davenport para VMO(NE\_D,k,p)

TEST	VALOR DEL TEST	VALOR DE LA DISTRIBUCIÓN	VALOR P	HIPÓTESIS
Friedman	9.571	15,507	0.296	Aceptada
Iman Davemport	1.214	2.032	0.297	Aceptada

En la tabla 19 podemos observar que el resultado del test no es mayor al valor de la distribución, es por esto que se acepta la hipótesis de semejanza entre los algoritmos probados, para este caso específicamente no se realiza test de comparaciones múltiples y se selecciona la variante VMO(NE\_D,12,5) como la de mayor valor medio (ver grafica 4.5).

Ilustración 7. Valores promedios de las pruebas para VMO(NE\_D,k,p)



#### 4.4.4 Obtención de la mejor variante para selección de Representatividad

Según los análisis realizado para el tipo de selección de representatividad, las variantes que mejores resultados obtuvieron fueron para el operador de unión

VMO con  $p=48$  y  $k=7$ , y para la unión de las diferencias fue VMO con  $p=12$  y  $k=5$ . En algunos casos obteniendo resultados significativamente mejores que las demás variantes y otros casos por alcanzar mayor valor medio. A continuación en la tabla 20 se presenta los resultados del test de Wilcoxon para comparar ambas variantes y seleccionar la que mejor comportamiento obtuvo para la selección elitista.

Tabla 20. Resultado del test de wilcoxon selección representativa

Comparación	R <sup>+</sup>	R <sup>-</sup>	Valor del test	Hipótesis
VMO(NE_U,48,7) vs VMO(NE_D,12,5)	174	58	0.02169	<b>Rechazada</b>

Claramente se puede apreciar que el valor del test es menor que el nivel de significancia determinado para el mismo ( $\alpha=0.05$ ) por lo que la hipótesis de igualdad es rechazada, es decir existe una diferencia significativa entre los dos algoritmos probados, para este caso específico necesitamos comparar los rangos positivos y negativos (ver capítulo II) que nos indica al algoritmo VMO(NE\_U,48,7) como la mejor variante para la selección representativa.

#### 4.4.5 Obtención de la mejor variante para el algoritmo VMO

Luego de realizar las pruebas a las distintas variantes implementadas del algoritmo VMO en el software Elvira, obtuvimos que la variante VMO(E\_U,24,3) para el tipo de selección elitista y VMO(NE\_U,48,7) para el de representatividad obtuvieron los mejores resultados, dentro de este apartado nos dedicaremos a obtener de los dos el mejor, aplicando el test de Wilcoxon obtuvimos los resultados mostrados en la tabla 21.

Tabla 21. Resultado del test de wilcoxon para obtener la mejor variante de VMO

Comparación	R <sup>+</sup>	R <sup>-</sup>	Valor del test	Hipótesis
VMO(E_U,24,3) vs VMO(NE_U,48,7)	115	138	0.35569	<b>Aceptada</b>

Claramente se puede apreciar que el valor del test es mayor que el nivel de significancia determinado, por lo que la hipótesis de igualdad es aceptada (columna hipótesis). Con estos resultados se puede concluir que ambas variantes presentan comportamientos similares, por lo que la selección de cualquiera de los dos queda totalmente justificada. En nuestro caso seleccionaremos la variante VMO(E\_U,24,3) ya que presenta mayor valor medio.

Luego de realizar el estudio de parámetros tomando en consideración la forma de selección (Elitista, Representativa) y el operador de contracción de la malla (Unión, Unión de Diferencias) podemos concluir que nos arrojó como resultado el algoritmo VMO con el tipo de selección Elitista, el operador unión y los parámetros  $p=24$  y  $k=3$  como la variante que mejores resultados obtuvo, no obstante no se desechan las variantes que arrojaron buenos resultados pues se la considerara para estudios posteriores en este ámbito.

#### 4.5 Análisis comparativo de VMO con el estado del arte

Del estudio de las variantes de VMO se obtuvo que la que mejor comportamiento presenta es VMO(E\_U,24,3). Seguidamente se presenta el estudio comparativo entre esta variante y los clasificadores bayesianos Naïve Bayes y Naïve Bayes aumentado a árbol (TAN) , RB K2, RB TAN y CBN descritos en la capítulo II, además de ByNet, BayesChaid, BayesPSO presentados en la tesis doctoral [2] y descritos en el marco referencial de este trabajo.

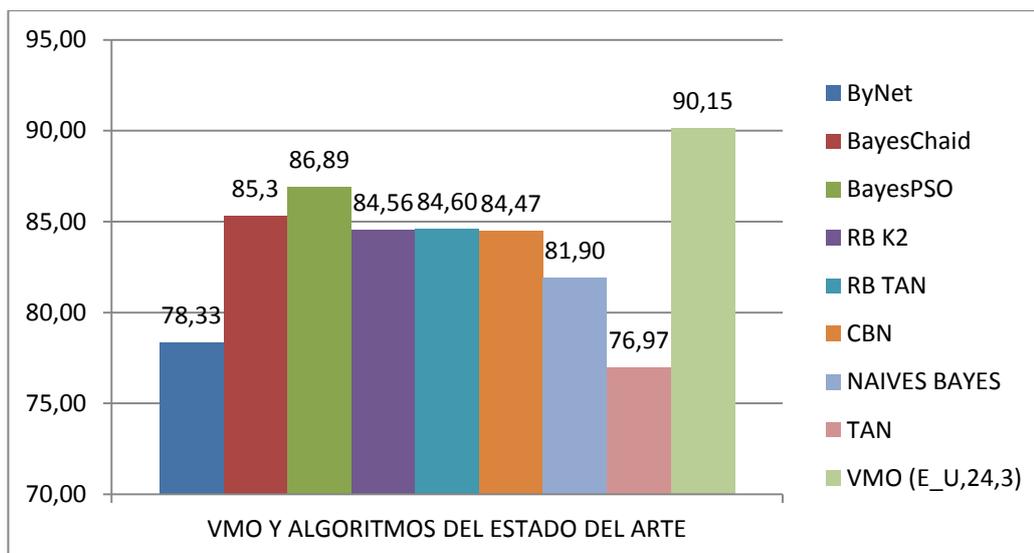
Los parámetros utilizados para los algoritmos ByNet, BayesChaid, BayesPSO, RB K2, RB TAN y CBN fueron los mismo que se presentan en su propuesta original detallada en [2]. La tabla 22 presenta los resultados del test de Friedman e Iman Davenport computado sobre los resultados mostrados en la tabla 28 del anexo 5, para saber si existen o no diferencias significativas en el grupo de algoritmos.

Tabla 22. Test de fridman e iman davenport vmo-estado del arte

TEST	VALOR DEL TEST	VALOR DE LA DISTRIBUCIÓN	VALOR P	HIPÓTESIS
Friedman	25.995	15,507	0.00105	Rechazada
Iman Davemport	3.929	2.032	4.316 E-4	Rechazada

La tabla 22 presenta los resultados de aplicar el test de Friedman e Iman Davenport, utilizando el valor de la distribución Chi Cuadrado con 8 grados de libertad para el test de Friedman, así como el valor de la distribución de Fisher con 8 y 104 grados de libertar para el test de Iman Davemport. Para ambos casos el valor del test es mayor que el valor de la distribución por lo que se rechaza la hipótesis de igualdad de medias y se concluye que existen diferencias significativas entre los algoritmos comparados.

Ilustración 8. Valores promedios de las pruebas VMO-estado del arte



La tabla 23 muestra los resultados del test de holm donde claramente se puede observar que su valor es menor que los presentados en la columna  $\alpha/i$ , es por esto que la hipótesis de semejanza es rechazada para todos los casos. Con este análisis se puede concluir que el algoritmo VMO (E\_U,24,3) obtiene resultados significativamente mejores que los que obtuvieron las demás variantes estudiadas en este apartado.

Tabla 23. Resultados del test de holm vmo-estado del arte

ALGORTIMO	z	Valor p	$\alpha/i$	Hipótesis
TAN	4.243	2.196E-5	0.0062	RECHAZADA
NAIVES BAYES	3.864	1.113E-4	0.0071	RECHAZADA
ByNet	3.795	1.474E-4	0.0083	RECHAZADA
CBN	2.725	0.0064	0.01	RECHAZADA
RB TAN	2.622	0.0087	0.0125	RECHAZADA
RB K2	2.415	0.0157	0.0166	RECHAZADA
BayesChaid	2.242	0.0243	0.025	RECHAZADA
BayesPSO	2.001	0.0453	0.05	RECHAZADA

Como pudimos observar en la tabla 23 el algoritmo de control (VMO(E\_U,24,3)) obtuvo resultados significativamente superiores sobre los todos los algoritmos

citados en dicha tabla, En este caso se selecciona el algoritmo de control VMO(E\_U,24,3). Ya que es significativamente mejor que todos los algoritmos comparados y alcanza mayor valor medio.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

## 5.1 Conclusiones

Luego de la finalización del presente trabajo investigativo desarrollado dentro del contexto de entrenamiento estructural de redes bayesianas y en base a la experiencia obtenida durante el transcurso de la misma se concluye lo siguiente:

- Se logró definir los operadores de exploración y contracción de la malla de soluciones, dentro de los cuales tenemos los operadores Unión y Unión de diferencias para el proceso de exploración y los operadores Elitista y Representativo para el de contracción de la malla.
- Se realizó un estudio de parámetros internos de VMO, el cual nos permitió obtener la variante de mejores resultados. En este proceso de pruebas se logró definir a la variante VMO con cantidad de nodos de malla inicial de 24 y el tamaño de la vecindad de 3, utilizando el operador de exploración Unión y el explorador de contracción Elitista, como la variante que obtuvo resultados significativamente mayores que las demás variantes estudiadas.
- Al obtener la mejor variante de VMO se la logró comparar con los algoritmos citados en el estado del arte [2], obteniendo resultados significativamente mayores y mejor promedio de clasificación con respecto a los algoritmos estudiados.

Con esto se puede concluir la siguiente investigación como satisfactoria, debido a que se obtuvo un modelo matemático basado en Optimización basada en mallas variables para el entrenamiento estructural de redes bayesianas.

## 5.2 Recomendaciones

Luego de la culminación del presente trabajo de investigación y en base a la experiencia obtenida del mismo se recomienda:

- Se continúe con el estudio de más variantes del algoritmo VMO para problemas de optimización discretos que no fueron consideradas en este trabajo investigativo debido a la falta de tiempo y lo extenso del tema.
- También se recomienda realizar la implementación del algoritmo VMO con búsqueda local y poder comparar con algoritmos de tipo clustering que existen en la comunidad científica.
- Y por último y no menos importante, se recomienda ampliar el presente estudio con estados del arte más actualizados, para comparar la efectividad del algoritmo VMO con respecto a nuevos algoritmos probados bajo el mismo ambiente.

## **CAPÍTULO VI**

## **BIBLIOGRAFÍA**

## 6.1 Literatura Citada

- [1] B. Pierre and B. Soren, *Bioinformatics: The Machine Learning Approach, Second Edition*. A Bradford Book, 2001.
- [2] M. Chávez. (2008) Monografias.com. [Online]. <http://www.monografias.com/trabajos-pdf2/modelos-redes-bayesianas-secuencias-genomicas/modelos-redes-bayesianas-secuencias-genomicas.pdf>
- [3] J. Pearl, "Probabilistic reasoning in intelligent systems," *Morgan Kaufmann*, pp. 11-13, 1988.
- [4] A. Puris, R. Bello, D. Molina, and F. Herrera, "Variable mesh optimization for continuous optimization problems," *Soft Computing*, pp. 511-525, 2012.
- [5] A. Asuncion and D. Newman. (2007) UCI Machine Learning Repository. [Online]. <http://archive.ics.uci.edu/ml/>
- [6] G. Cooper-Herskovits and E. H. Herskovits, "A Bayesian methods for the induction of probabilistic networks from data.," *Machine Learning*, pp. 309-348, 1992.
- [7] P. Larrañaga, M. Poza, Y. Yurramendi, and R. H. Murga, "Structure Learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters," *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 912-926, 1996.
- [8] M. L. Wong and K. S. Leung, "An efficient data mining method for learning Bayesian networks using an evolutionary algorithm-based hybrid approach," *Evolutionary Computation, In: Transactions on*, pp. 378-404, 2004.
- [9] Definicion.de. (2008) Definición.de. [Online]. <http://definicion.de/>
- [10] e. eumednet. (2013) eumednet. [Online]. <http://www.eumed.net/tesis-doctorales/2007/cavl/Grafos%20aciclicos%20dirigidos.htm>
- [11] P. Felgaer, "Optimización de Redes Bayesianas basado en Técnicas de Aprendizaje por Inducción ," *Reportes Técnicos en Ingeniería del Software.*, pp. 64-69, 2004.
- [12] W. Fritz, R. García-Martínez, A. Rama, and Blanqué, "The Autonomous

Intelligent System," *Elsevier Science Publishers*, pp. 109-125, 1989.

- [13] R. García-Martínez and D. Borrajo, "An Integrated Approach of Learning, Planning and Executing," *Journal of Intelligent and Robotic Systems*, pp. 47-78, 2000.
- [14] B. Gilles and P. Bratle, *Fundamentos de Algoritmos*. Madrid: Prentice Hall, 1997.
- [15] H. A. Simon, *Why should machines learn?*. Palo Alto CA: Tioga. : Michalski et al., 1983.
- [16] R. O. Duda and P. E. Hart, "Pattern Classification and scene analysis," *Jonh Wiley Sons*, pp. 23-28, 1973.
- [17] N. Friedman and M. Goldszmidt, "Building Classifiers using Bayesian Networks," in *Proceedings of Thirteen National Conference on Artificial Intelligence*, 1996, pp. 1277-1284.
- [18] P. Larrañaga, "Aprendizaje automatico de Modelos Graficos II Aplicaciones a la Clasificación Supervisada," *Sistemas expertos probabilísticos*, pp. 141-162, 2000.
- [19] C. Chow-Liu and C. Liu, "Approximating discrete probability distribution with dependence trees," *IEEE Transactions on Information Theory*, pp. 462-467, 1968.
- [20] G. Rebane-Pearl and J. Pearl, "The recovery of causal poly- trees from statistical data," *Int. J. Approx. Reasoning*, p. 341, 1988.
- [21] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers," *MACHINE LEARNING*, pp. 131-163, 1997a.
- [22] F. V. Jensen and T. D. Nielsen, "Bayesian Networks and Decisions Graphs," *Information Science and Statistics Series, Springer Verlag, New York*, 2007.
- [23] M. Sahami, "Learning limited dependence Bayesian Classifiers," *In Proceeding of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 335-338, 1996.
- [24] W. Dillon and M. Goldstein, "Multivariate Analysis," *Methods and Aplicacions. John Wiley & Sons.*, 1984.

- [25] B. Escofier and J. Pages, "Análisis Factoriales Simples y Múltiples," *Universidad del País Vasco. Bilbao*, 1992.
- [26] M. Lebart, "Statistique Exploratoire Multidimensionnelle," *Dunod. París*, 1998.
- [27] D. Peña, "Análisis de Datos Multivariantes," *MacGraw Hill*, p. 556, 2002.
- [28] R. Ruiz, "Heurísticas de selección de atributos para datos de gran dimensionalidad," *Tesis presentada en opción al grado de Doctor en Informática*, 2006.
- [29] G. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," *In Machine Learning: Proceeding of Eleventh International Conference, Morgan Kaufman*, 1994.
- [30] J. Brender, J. Talmon, and M. Egmont-Petersen, "Measuring quality of medical knowledge.," *Medical Informatics in Europe Lisbon*, 1994.
- [31] C. Van Rijsbergen, *Information Retrieval*. London, Butterworths, 1979.
- [32] J. Ruiz-Shulcloper, "Logical Combinatorial Pattern Recognition," 2000.
- [33] M. Donald, C. Spiegelhalter, and J. Taylor, "Machine learning, neural and statistical classification," *Ellis Horwood Limited*, p. 289, 1994.
- [34] A. Puris, *Desarrollo de meta-heurísticas poblacionales para la solución de problemas complejos*. Cuba: Universidad Central "Marta Abreu" de Las Villas, 2009.
- [35] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal Machine Learn* , pp. 1-30, 2006.
- [36] S. García and F. Herrera, "An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons," *Journal of Machine Learning Research*, pp. 2677-2694, 2008.
- [37] D. Sheskin, "Handbook of parametric and nonparametric statistical procedures," *CRC, Editor Chapman & Hall: London/West Palm Beach.*, 2006.
- [38] S. García, "A study of statistical techniques and performance measures for geneticsbased machine learning: accuracy and interpretability," *Soft Computing and Applications*, pp. 959-977, 2009.

- [39] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the Friedman statistic.," *Commun Stat*, pp. 571-595, 1980.
- [40] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand Journal Stat*, pp. 65-70, 1979.
- [41] F. Wilcoxon, "Adjusted p-values for simultaneous inference," *Biometrics*, pp. 1005-1013, 1945.
- [42] J. Zar, "Biostatistical analysis," *Prentice Hall, Englewood Cliffs*, 1999.
- [43] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, pp. 675-701, 1937.
- [44] M. Friedman, "A Comparison of Alternative Tests of Significance for the Problem of m Ranking," *Annals of Mathematical Statistics*, pp. 86-92, 1940.
- [45] W. L. Buntine, "Operation for Learning with Graphical Models," *Artificial Intelligence Research*, pp. 159-225, 1994.
- [46] W. L. Buntine, "1995," *Graphical Models for Discovery Knowledge and Data Mining*, pp. 2-11, 1995.
- [47] D. Heckerman, *A Tutorial on Learning with Bayesian Networks*. MIT Press Cambridge, MA, USA: M. Jordan, 1999.
- [48] A. G. Hernández, "Aprendizaje Automático: Árboles de Decisión.," *Aprendizaje BAYesiano*, pp. 24-29, 2004.
- [49] H. Aytug, "Decision Tree Induction," *University of Florida.*, 2000.
- [50] N. Friedman, M. Goldszmidt, D. Heckerman, and Russ, "Challenge: Where is the impact of Bayesian networks in learning?," *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence 1*, 1997b.
- [51] M. J. Pazani, "Searching for dependences in Bayesian classifiers," *Learning from data: Artificial Intelligence. Proceeding of the Twelfth Conference, Horvitz, E. Jensen, F. (eds), Morgan Kaufman*, pp. 414-419, 1996.
- [52] P. Larrañaga, I. Inza, and A. Moujahid, "Modelos Probabilísticos para la Inteligencia Artificial y la Minería de Datos: Selección de Variables," *Curso de Doctorado*, 2003.

- [53] P. Lanzi, "Feature Subset Selection Using Effective Combine of Filter and Wrapper Approaches," *Tesis de Grado*, 2006.
- [54] V. C. Daalen, "Evaluating Medical Knowledge Based Systems," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 3, pp. 888-889, 1992.
- [55] C. Torres, *METODOLOGÍA DE LA INVESTIGACIÓN*. Pearson Educación., 2010.

## **CAPÍTULO VII**

### **ANEXOS**

## Anexo 1

Tabla 24. Resultados obtenidos para VMO(E\_U,k,p)

<b>Bases del Conocimiento</b>	<b>VMO (E_U,12,3)</b>	<b>VMO (E_U,12,5)</b>	<b>VMO (E_U,12,7)</b>	<b>VMO (E_U,24,3)</b>	<b>VMO (E_U,24,5)</b>	<b>VMO (E_U,24,7)</b>	<b>VMO (E_U,48,3)</b>	<b>VMO (E_U,48,5)</b>	<b>VMO (E_U,48,7)</b>
mammographic	84,18	84,29	84,50	84,39	84,50	84,18	83,87	83,77	83,77
Lung-cancer	75,00	78,13	75,00	78,13	78,13	75,00	68,75	68,75	75,00
Hepatitis	92,90	92,90	92,90	93,55	93,55	92,26	90,97	90,97	90,97
e colic	81,85	77,08	76,19	76,49	76,79	76,79	77,68	76,19	76,79
breast-cancer-w	97,85	98,14	98,00	97,85	98,00	97,57	97,57	97,57	97,71
contac-lenses	91,67	95,83	95,83	95,83	95,83	87,50	87,50	87,50	95,83
hayes-roth-m	83,33	84,09	83,33	84,09	84,09	75,00	79,55	80,30	83,33
Monk 1	90,80	89,52	89,52	97,58	89,52	91,94	87,90	94,35	95,97
Vote	93,10	93,33	92,87	93,79	93,33	92,41	92,87	92,00	92,87
Balance-scale	86,08	86,08	86,88	86,72	86,40	84,96	84,48	85,76	84,64
tic-tac-toe	74,84	75,68	75,99	75,68	75,47	73,49	72,76	74,01	74,01
Iris	98,67	98,67	97,33	98,00	98,00	96,67	96,67	96,67	96,67
Labor	100,00	100,00	100,00	100,00	100,00	95,00	97,50	95,00	97,50
soybean	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

## Anexo 2

Tabla 25. Resultados obtenidos para VMO(E\_D,k,p)

<b>Bases del Conocimiento</b>	<b>VMO (E_D,12,3)</b>	<b>VMO (E_D,12,5)</b>	<b>VMO (E_D,12,7)</b>	<b>VMO (E_D,24,3)</b>	<b>VMO (E_D,24,5)</b>	<b>VMO (E_D,24,7)</b>	<b>VMO (E_D,48,3)</b>	<b>VMO (E_D,48,5)</b>	<b>VMO (E_D,48,7)</b>
mammographic	84,18	84,29	84,60	84,39	84,39	84,70	84,29	84,39	84,60
Lung-cancer	75,00	75,00	75,00	78,13	75,00	75,00	75,00	75,00	75,00
hepatitis	92,90	93,55	92,90	92,90	93,55	92,90	92,90	92,90	92,90
e colic	77,08	77,38	76,49	77,08	77,08	76,49	76,49	77,38	77,08
breast-cancer-w	97,85	97,85	98,00	97,85	97,85	98,00	97,85	97,85	98,00
contac-lenses	91,67	95,83	95,83	91,67	95,83	95,83	91,67	95,83	95,83
hayes-roth-m	82,58	82,58	84,09	84,85	84,09	82,58	82,58	84,85	84,09
Monk 1	91,13	90,32	89,52	90,32	90,32	87,90	90,32	90,32	87,90
vote	93,33	93,10	93,10	93,33	92,87	92,87	93,33	92,87	93,10
Balance-scale	86,88	86,56	86,40	86,56	85,76	86,08	86,08	85,76	85,92
tic-tac-toe	74,95	75,47	75,05	75,26	75,05	74,84	75,26	75,05	75,00
iris	97,33	97,33	98,00	97,33	97,33	98,00	97,33	97,33	98,00
labor	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00
soybean	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

### Anexo 3

Tabla 26. Resultados obtenidos para VMO(NE\_U,k,p)

Bases del Conocimiento	VMO (NE_U,12,3)	VMO (NE_U,12,5)	VMO (NE_U,12,7)	VMO (NE_U,24,3)	VMO (NE_U,24,5)	VMO (NE_U,24,7)	VMO (NE_U,48,3)	VMO (NE_U,48,5)	VMO (NE_U,48,7)
mammographic	84,29	83,45	83,35	83,35	83,35	83,56	83,87	83,77	83,77
Lung-cancer	75	62,5	68,75	68,75	65,63	78,13	68,75	68,75	75,00
hepatitis	92,9	91,61	90,32	92,26	91,61	92,26	90,97	90,97	90,97
e colic	81,55	76,19	75,3	75,6	76,79	76,79	77,68	76,19	76,79
breast-cancer-w	97,85	97,42	97,57	97,57	97,85	97,57	97,57	97,57	97,71
contac-lenses	91,67	87,5	83,33	87,5	91,67	87,5	87,5	87,5	95,83
hayes-roth-m	81,82	79,55	78,79	75	79,55	75	79,55	80,3	83,33
Monk 1	88,71	80,65	86,29	84,68	90,32	91,94	87,9	94,35	95,97
vote	93,1	92,41	92,64	92,41	92,41	92,41	92,87	92,64	92,87
Balance-scale	86,24	85,28	83,68	85,76	84	84,96	84,48	85,76	84,64
tic-tac-toe	75,26	73,28	72,86	74,01	74,43	73,49	72,76	74,01	74,01
iris	96,67	96,67	96,67	97,33	96,67	96,67	96,67	96,67	96,67
labor	100	95	97,5	97,5	95	95	97,5	95	97,50
soybean	100	100	100	100	100	100	100	100	100,00

## Anexo 4

Tabla 27. Resultados obtenidos para VMO(NE\_D,K,P)

Bases Del Conocimiento	VMO (NE_D,12,3)	VMO (NE_D,12,5)	VMO (NE_D,12,7)	VMO (NE_D,24,3)	VMO (NE_D,24,5)	VMO (NE_D,24,7)	VMO (NE_D,48,3)	VMO (NE_D,48,5)	VMO (NE_D,48,7)
mammographic	83,77	83,35	83,56	83,14	84,18	83,25	83,77	83,56	83,77
Lung-cancer	68,75	71,88	65,63	65,63	65,63	62,5	65,63	68,75	68,75
hepatitis	89,68	90,97	90,97	90,32	90,97	89,68	90,91	90,18	90,97
e colic	75,89	75,30	75,3	76,19	75,3	75,3	75,89	75,3	76,2
breast-cancer-w	97,57	97,57	97,71	97,57	97,71	97,57	97,57	97,71	97,71
contac-lenses	83,33	91,67	83,33	87,5	91,67	83,33	87,5	91,67	83,33
hayes-roth-m	71,97	81,06	78,79	78,79	78,79	81,06	78,79	78,79	78,79
Monk 1	80,65	89,52	84,68	87,1	85,16	84,68	88,71	86,29	84,68
vote	92,41	91,72	93,1	92,41	92,41	93,1	92,41	92,41	91,72
Balance-scale	84,16	84,96	83,36	83,68	84,96	83,36	83,68	84,96	84,16
tic-tac-toe	73,07	73,80	73,9	73,8	75,37	73,07	73,59	73,49	73,28
iris	98	97,33	96,67	96,67	97,33	96,67	96,67	97,33	96,67
labor	95	95,00	100	95	95	100	100	100	95
soybean	100	100,00	100	100	100	100	100	100	100

## Anexo 5

Tabla 28. Resultados obtenidos para VMO-estado del arte

Bases del conocimiento	ByNet	BayesChaid	BayesPSO	RB K2	RB TAN	CBN	NAIVES BAYES	TAN	VMO (E_U,24,3)
mammographic	81,89	83,14	82,62	82,41	81,27	83,25	81,99	79,29	84,39
Lung-cancer	78,13	75,00	78,13	71,88	65,63	75,00	56,25	46,87	78,13
hepatitis	85,16	86,45	83,87	84,52	85,16	85,16	87,09	87,74	93,55
e colic	67,26	85,12	84,52	85,12	84,82	85,42	77,38	62,79	76,49
breast-cancer-w	90,78	97,51	97,36	97,36	95,31	97,36	97,13	87,83	97,85
contac-lenses	87,50	83,33	83,33	70,83	66,67	70,83	83,33	66,66	95,83
hayes-roth-m	57,58	81,06	74,24	72,73	67,42	80,30	66,66	57,57	84,09
Monk 1	72,58	70,97	99,19	79,03	95,97	77,42	73,38	57,25	97,58
vote	94,33	91,67	92,67	91,33	93,67	89,67	89,65	94,03	93,79
Balance-scale	63,52	92,16	93,92	92,16	92,96	92,16	86,40	82,56	86,72
tic-tac-toe	70,25	72,96	72,65	76,62	76,83	69,62	70,98	73,06	75,68
iris	95,33	94,00	95,33	94,00	94,67	94,00	91,33	92,00	98,00
labor	84,21	87,72	89,47	91,22	89,47	89,47	85,00	90,00	100,00
soybean	68,08	93,11	89,16	94,58	94,58	92,97	100,00	99,87	100,00

## Anexo 6 Matriz de relación entre la sistematización, los objetivos y los resultados.

SISTEMATIZACION	OBJETIVOS	RESULTADOS
¿Cómo redefinir los operadores de exploración y contracción de la Meta-heurística Optimización basada en Mallas Variables para el problema discreto de estudio?	Definir los operadores de exploración y contracción de la meta-heurística.	Se obtuvo un modelo matemático que nos permita un aprendizaje estructural lo más viable posible.
¿Cómo Realizar un estudio de parámetros satisfactorio de manera que se pueda obtener la mejor de las variantes implementadas?	Realizar un estudio de parámetros indicando cuál de las posibles variantes sea la de mejor comportamiento.	Se realizó un estudio de parámetros basados en la selección de la malla inicial y los operadores de expansión, teniendo como resultado que la mejor variante es VMO(E_U,24,3).
¿De qué manera se puede validar la calidad de los resultados que se obtendrán de la Meta-heurística Optimización Basada en Mallas Variables?	Comparar los resultados alcanzados por la propuesta con otros algoritmos propuestos en documentos citados.	Se Aplicó nuestra propuesta a diferentes bases de conocimientos internacionales y obtuvimos resultados competitivos frente a los algoritmos propuestos en el estado del arte.