



# **UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**

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

### **CARRERA INGENIERÍA EN SISTEMAS**

Proyecto de Investigación  
previo a la obtención de título  
de Ingeniero en Sistemas.

#### **Título del Proyecto de Investigación:**

**“APRENDIZAJE ESTRUCTURAL DE REDES BAYESIANAS EN ENTORNOS  
LOCALES A TRAVÉS DE UNA MALLA VARIABLE”**

#### **Autor:**

Irina Monserrat Pacheco Zamora

#### **Director de Proyecto de Investigación:**

PhD. Msc. Amilkar Puris Cáceres

Quevedo – Ecuador.

2015

## DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **Irina Monserrat Pacheco 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.

f. \_\_\_\_\_

Irina Monserrat Pacheco Zamora

## **CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN**

El suscrito, Amilkar Yudier Puris Cáceres, Docente de la Universidad Técnica Estatal de Quevedo, certifica que el estudiante Irina Monserrat Pacheco Zamora, realizó el Proyecto de Investigación de grado titulado “**APRENDIZAJE ESTRUCTURAL DE REDES BAYESIANAS EN ENTORNOS LOCALES A TRAVÉS DE UNA MALLA VARIABLE**”, previo a la obtención del título de Ingeniería en Sistemas, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

.....

Phd. Msc. Amilkar Puris Cáceres

**DIRECTOR DE PROYECTO DE INVESTIGACIÓN**



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**FACULTAD DE CIENCIAS DE LA INGENIERÍA**  
**CARRERA DE INGENIERÍA EN SISTEMAS**

**PROYECTO DE INVESTIGACIÓN**

**Título:**

“Aprendizaje estructural de redes bayesianas en entornos locales a través de una malla variable”

Presentado a la Comisión Académica como requisito previo a la obtención del título de Ingeniero en Sistemas.

Aprobado por:

---

PRESIDENTE DEL TRIBUNAL DE TESIS

Ing. Byron Oviedo Bayas.

---

MIEMBRO DEL TRIBUNAL DE TESIS

Ing. Iván Jaramillo Chuqui.

---

MIEMBRO DEL TRIBUNAL DE TESIS

Ing. Pavel Novoa Hernández.

QUEVEDO – LOS RÍOS - ECUADOR.

2015

## AGRADECIMIENTOS

*Agradezco a Dios por su infinito amor, su bondad, sus bendiciones, por la maravillosa familia que me ha otorgado, por enseñarme cada día que confiando en Él no hay ningún obstáculo que no se pueda superar.*

*A mis padres José Eduardo Benjamín e Irina Jakeline por haberme apoyado en todo momento, por sus consejos, sus valores, pero más que nada por su infinito amor.*

*A mi hermano José Andrés por la confianza, el cariño, la amistad y el apoyo que siempre me ha brindado.*

*A mi gran familia en general por el cariño que han demostrado cada día de mi vida.*

*Al Dr. Amilkar Puris Cáceres por compartir desinteresadamente sus conocimientos, su experiencia y paciencia para la realización del presente proyecto de investigación.*

*A mi amiga Cinthya y a mi primo Luis Enrique que siempre los he considerados como hermanos, gracias por el apoyo, la ayuda, los ánimos y su compañía en cada momento.*

## DEDICATORIA

*A Dios.*

*A mis padres José Eduardo e Irina Jakeline.*

*A mi hermano José Andrés.*

## RESUMEN EJECUTIVO

En este trabajo de investigación se presenta una propuesta de incorporar búsqueda local al modelo de Optimización basado en Mallas Variables (VMO). En esta propuesta se aplicarán tres algoritmos de Búsqueda en Entorno Variable (VNS) que será la encargada de explotar las regiones cercanas a las buenas soluciones ya encontradas con el objetivo de mejorar el proceso de búsqueda y obtener mejores soluciones que solamente utilizando el algoritmo VMO para el entrenamiento estructural de Redes Bayesianas.

Una vez seleccionada la hibridación del VMO con búsqueda local que mejor resultados obtuvieron con los parámetros, intensidad y métrica establecidos más adelante en este proyecto, serán comparados con los resultados de otros algoritmos que se encuentran en el artículo “Un método híbrido para el aprendizaje de redes bayesianas basado en la optimización de colonia de hormigas”.

Para el desarrollo y ejecución del algoritmo VMO con búsqueda local fueron utilizadas algunas de las clases principales del software “Entorno para el desarrollo de modelos gráficos probabilísticos (Elvira)”, el cual fue seleccionado debido a su capacidad para la construcción de Redes Bayesianas y ser una herramienta de código abierto.

Palabras claves: Redes Bayesianas, Optimización basado en Mallas Variables (VMO), Búsqueda en Entorno Variable (VNS).

## **ABSTRACT AND KEYWORDS**

This research paper presents a proposal to incorporate local search into the model of Optimization based on Variable Meshes (VMO). In this proposal, three algorithms of search will be applied in Variable Environment (VNS) which will be in charge of exploiting the close regions to good solutions already found in order to improve the search process and to obtain better solutions that only using the VMO algorithm for structural training of Bayesian Networks.

Once selected the hybridization of the VMO with local search to better results were obtained with the parameters, intensity and metric set out later in this project, will be compared with the results of other algorithms that are found in the article "A hybrid method for learning bayesian networks based on ant colony optimization".

For the development and execution of the VMO algorithm with local search, were used some of the main classes of software "Environment for the development of graphical models probabilistic (Elvira)", which was selected because of its ability for the construction of Bayesian Networks and be an open source tool.

**Keywords:** Bayesian Networks, Optimization based on Variable Meshes (VMO), Variable Neighbourhood Search (VNS).

# ÍNDICE

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS .....	ii
CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN.....	iii
AGRADECIMIENTOS.....	v
DEDICATORIA.....	vi
RESUMEN EJECUTIVO .....	vii
ABSTRACT AND KEYWORDS .....	viii
CÓDIGO DUBLÍN .....	xiv
INTRODUCCIÓN.....	1
CAPÍTULO I.....	3
CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN .....	3
1.1. Problema de investigación. ....	4
1.1.1. Planteamiento del problema. ....	4
1.1.2. Formulación del problema.....	4
1.1.3. Sistematización del problema.....	5
1.2. Objetivos.....	6
1.2.1. Objetivo General.....	6
1.2.2. Objetivos Específicos. ....	6
1.3. Justificación. ....	7
CAPÍTULO II.....	8
FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN .....	8
2.1. Marco Conceptual.....	9
2.1.1. Redes bayesianas. ....	9
2.1.2. Aprendizaje de Redes Bayesianas. ....	9
2.1.3. Metaheurísticas Poblacionales.....	11
2.1.4. Optimización Basada en Mallas Variables (VMO).....	11

2.1.5.	Búsqueda Local. ....	15
2.1.6.	Metaheurísticas poblaciones con búsqueda local. ....	16
2.1.7.	Búsqueda en Entorno Variable en el espacio de Grafos Dirigidos Acíclicos.....	17
2.1.8.	Métrica k2.....	17
2.1.9.	ACO-B.....	18
2.1.10.	Test de Friedman .....	19
CAPÍTULO II.....		20
METODOLOGÍA DE LA INVESTIGACIÓN.....		20
3.1.	Localización.....	21
3.2.	Tipo de Investigación.....	21
3.2.1.	Exploratoria .....	21
3.3.	Métodos de Investigación. ....	21
3.3.1.	Analítico .....	21
3.4.	Fuentes de recopilación de información. ....	21
3.5.	Diseño de Investigación.....	21
3.5.1.	Diseño Cuasi – Experimental. ....	21
3.5.2.	Pasos de un cuasi experimento.....	22
3.6.	Instrumentos de Investigación. ....	22
3.7.	Tratamientos de los datos. ....	22
3.8.	Recursos humanos y materiales.....	23
3.8.1.	Hardware. ....	23
3.8.2.	Software.....	23
3.8.3.	Suministros.....	24
3.8.4.	Personal. ....	24
CAPÍTULO IV .....		25
RESULTADOS Y DISCUSIÓN .....		25

1.4. Resultados.....	26
4.1.1. Estudio experimental de la aplicación de búsqueda local al algoritmo VMO.....	27
4.1.2. Integración del algoritmo VMO con búsqueda local en el software Elvira. .	27
4.1.3. Análisis de los resultados. ....	29
4.1.4. Análisis comparativo del algoritmo VMO con búsqueda local con el estado del arte.....	32
CAPÍTULO V .....	34
CONCLUSIONES Y RECOMENDACIONES .....	34
5.1. Conclusiones.....	35
5.2. Recomendaciones. ....	36
CAPÍTULO VI.....	37
6.1. Bibliografía. ....	38
CAPÍTULO VII.....	40
ANEXOS.....	40

## ÍNDICE DE ILUSTRACIONES

Ilustración 2. 1. Ejemplo de una Red Bayesiana.....	9
Ilustración 2. 2. Espacio de búsqueda .....	16
Ilustración 2. 3 Pseudocódigo del algoritmo VNSST .....	17
Ilustración 2. 4. Pseudocódigo del algoritmo ACO-B.....	18

## ÍNDICE DE TABLAS

Tabla 1. Materiales de hardware. ....	23
Tabla 2. Materiales de software. ....	23
Tabla 3. Tabla de Suministro. ....	24
Tabla 4. Personal. ....	24
Tabla 5. Resultados con una intensidad de 150 iteraciones. ....	30
Tabla 6. Resultados con una intensidad de 250 iteraciones ....	30
Tabla 7. Resultados con una intensidad de 350 iteraciones ....	30
Tabla 8. Test De Friedman E Iman Davenport VMO-Búsqueda Local.....	31
Tabla 9. Resultados de la comparación entre la hibridación de Búsqueda con el Algoritmo VMO y los algoritmos citados en el estado del arte. ....	32
Tabla 10. Test de Friedman e Iman Davenport VMO+VNSSTPC - estado del arte.....	33

## CÓDIGO DUBLÍN

Título:	Aprendizaje estructural de redes bayesianas en entornos locales a través de una malla variable.
Autor:	Irina Monserrat Pacheco Zamora.
Palabras Claves:	Aprendizaje Estructural, Redes Bayesianas, VMO, Mallas Variables, Búsqueda Local, VNS.
Resumen:	<p><b>Resumen.-</b> En este trabajo de investigación se presenta una propuesta de incorporar búsqueda local al modelo de optimización basado en mallas variables (VMO). En esta propuesta se aplicarán tres algoritmos de búsqueda en entorno variable (VNS) que será la encargada de explotar las regiones cercanas a las buenas soluciones ya encontradas con el objetivo de mejorar el proceso de búsqueda y obtener mejores soluciones que solamente utilizando el algoritmo VMO para el entrenamiento estructural de redes bayesianas.</p> <p><b>Abstract.-</b> This research paper presents a proposal to incorporate local search into the model of Optimization based on Variable Meshes (VMO). In this proposal, three algorithms of search will be applied in Variable Environment (VNS) which will be in charge of exploiting the close regions to good solutions already found in order to improve the search process and to obtain better solutions that only using the VMO algorithm for structural training of Bayesian Networks.</p>
Descripción:	54 hojas : dimensiones, 29 x 21 cm + CD-ROM 6162
URI:	

## INTRODUCCIÓN

Las redes bayesianas [1] son importantes modelos probabilísticos dentro del campo de la inteligencia artificial, dado que han obtenido excelentes resultados en el manejo de grandes volúmenes de datos, ya que relacionan un conjunto de variables aleatorias mediante un grafo dirigido acíclico en el que cada nodo es una variable y cada enlace una dependencia probabilística. Las mismas proveen una forma compacta de representar el conocimiento y métodos flexibles de razonamiento capaces de predecir el valor de variables no observadas y explicar las observadas.

La obtención de una red bayesiana a partir de datos es un proceso de aprendizaje que se divide en dos etapas: el aprendizaje estructural y el aprendizaje paramétrico. La primera de ellas consiste en obtener la estructura de la red bayesiana, es decir, las relaciones de dependencia e independencia entre las variables involucradas. La segunda etapa, tiene como finalidad obtener las probabilidades a priori y condicionales requeridas, a partir de una estructura dada.

Este proyecto de investigación hace referencia al aprendizaje estructural de redes bayesianas en entorno locales a través de mallas variables. El modelo Optimización basada en Mallas Variable (VMO) [2] [3] es una metaheurística poblacional [4], que de acuerdo con [5] su esencia radica en la representación de soluciones potenciales como nodos de una malla, cuya configuración varía en su movimiento por el espacio de búsqueda.

Conforme a lo publicado en [6] una metaheurística es un modelo de amplio propósito que guía heurísticas específicas hacia regiones promisorias del espacio de búsqueda. Mientras que [7] manifiesta que los algoritmos de búsqueda local se encargan de intensificar la búsqueda en las cercanías de las zonas exploradas.

La hibridación de metaheurísticas poblacionales con los métodos de búsqueda local permite obtener un balance adecuado entre la exploración y explotación, además de que la incorporación de búsqueda local a los algoritmos poblacionales puede alcanzar mejores resultados que solo utilizando una metaheurística poblacional.

Para la realización de la investigación se utiliza el software llamado Entorno para el desarrollo de modelos gráficos probabilísticos (Elvira) que es el fruto de un proyecto de investigación, en el que participan investigadores de varias universidades españolas y de otros centros. Elvira tiene como objetivo la construcción, edición y evaluación de redes bayesianas.

Para la ejecución del siguiente proyecto de investigación se utilizó la base de conocimiento denominada “*alarm*” que se encuentra en Bayesian Network Repository [8], para luego realizar un estudio comparativo entre la variante del modelo basado en mallas variables con búsqueda local con algunos algoritmos del estado del arte.

## **CAPÍTULO I**

### **CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN**

## **1.1. Problema de investigación.**

### **1.1.1. Planteamiento del problema.**

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. En la investigación realizada por [9] se desarrolló un estudio para definir otra forma de aprendizaje estructural basado en la búsqueda de estructuras a través de la optimización basada en mallas variables (VMO) que es una metaheurística poblacional que su esencia radica en la representación de soluciones potenciales como nodos de una malla, cuya configuración varía en su movimiento por el espacio de búsqueda.

Cabe mencionar, que en la investigación [9] para el desarrollo de la estructura del algoritmo VMO para el problema del aprendizaje estructural de redes bayesianas en ambientes discretos, se utilizó la métrica calidad de clasificación. Esta métrica (calidad de la clasificación) incorpora un alto grado de complejidad a la búsqueda debido al tiempo que se demora en aplicar la clasificación a una determinada red.

Además, basándose en [7] donde se manifiesta que la incorporación de algoritmos de búsqueda local a las metaheurísticas poblacionales puede alcanzar mejores resultados que solo utilizando algoritmos poblacionales o búsqueda local de manera separada, se ve en la necesidad de aplicar búsqueda local al algoritmo VMO desarrollado en [9], utilizando la métrica  $k_2$  que a diferencia de la métrica calidad de clasificación realiza una búsqueda voraz y muy eficaz para encontrar una red de buena calidad en un tiempo aceptable y así mejorar el proceso de búsqueda para encontrar soluciones de calidad.

### **1.1.2. Formulación del problema.**

¿Cómo adaptar la metaheurística basada en mallas variables para el entrenamiento de redes bayesianas?

### **1.1.3. Sistematización del problema.**

- ¿Será posible mejorar el comportamiento del algoritmo VMO incorporándole búsqueda local?
- ¿Qué valor de intensidad proporciona un buen desempeño a la hibridación de la búsqueda local con el algoritmo VMO?
- ¿Cómo evaluar la calidad de los resultados que se obtendrán con la incorporación de búsqueda local al algoritmo VMO?

## **1.2. Objetivos.**

### **1.2.1. Objetivo General.**

Obtener una hibridación entre el algoritmo basado en mallas variables con búsqueda local para el entrenamiento de redes bayesianas.

### **1.2.2. Objetivos Específicos.**

- Definir la intensidad de la búsqueda local.
- Seleccionar el algoritmo de búsqueda local que mejor resultados obtuvo.
- Realizar un estudio comparativo entre el algoritmo propuesto y otros algoritmos del estado del arte.

### **1.3. Justificación.**

En la actualidad el uso combinado de los algoritmos poblacionales con la búsqueda local ha demostrado obtener mejores resultados, que solo utilizando uno de los dos algoritmos de manera separada, es por eso que se han venido desarrollando algoritmos que buscan abordar el problema de la búsqueda, es decir que la búsqueda global tenga la capacidad de alcanzar soluciones rápidamente aprovechando las ventajas de la búsqueda local. A estos algoritmos que combinan ambos componentes (algoritmo poblacional con búsqueda local) se los denomina: Algoritmos Híbridos, Algoritmos Genéticos Locales, etc [7].

El objetivo de la hibridación es hacer que la búsqueda local y la metaheurística poblacional trabajen de forma cooperativa para conseguir una sinergia entre ambos que permita mejorar el proceso de búsqueda. Numerosos han sido los proyectos que combinan ambos componentes tales como el Algoritmo híbrido con múltiples colonias (ACOR), Algoritmo basado en encadenamiento de búsquedas locales para problemas de optimización continua y el AMCR-BLC<sup>2</sup> P.

Con lo anteriormente mencionado, el objetivo del presente proyecto de investigación tiene como finalidad aplicar búsqueda local al algoritmo VMO para el entrenamiento de redes bayesianas. De esta forma explotar las regiones cercanas a las buenas soluciones ya encontradas, y así obtener soluciones de mejor calidad que solo utilizando uno de los dos algoritmos.

## **CAPÍTULO II**

# **FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN**

## 2.1. Marco Conceptual.

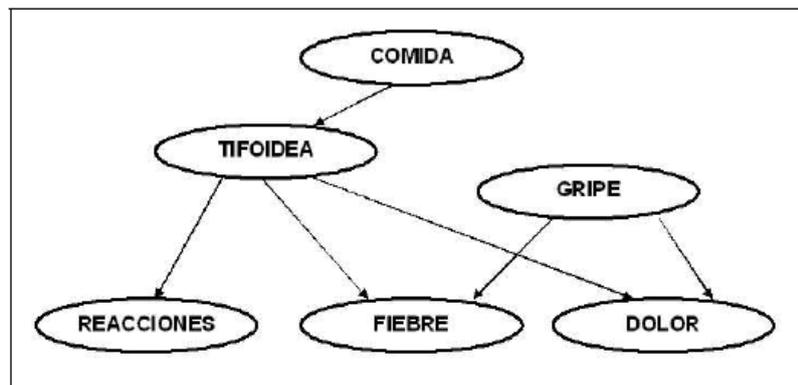
### 2.1.1. Redes bayesianas.

Las Redes Bayesianas (RB) constituyen un formalismo muy atractivo de representación del conocimiento, resultado de la sinergia entre métodos probabilísticos-estadísticos de análisis de datos y técnicas de Inteligencia Artificial [10].

Una RB es un par  $(D, P)$ , donde  $D$  es un grafo acíclico dirigido,  $P(x) = \prod_{i=1}^n P(x_i | pa(x_i))$  un conjunto de  $n$  distribuciones de probabilidad condicionales, una por cada variable  $x_i$  (nodos del grafo), y  $pa(x_i)$  es el conjunto de padres del nodo  $x_i$  en  $D$  [11].

Las redes bayesianas también son conocidas como redes de creencia bayesianas; según [12] [13], se definen como un grafo acíclico dirigido, también se pueden referenciar como redes causales [14] [15], redes de creencia, modelos recursivos, redes probabilistas, aunque se puede asumir el término RB [16].

En la Ilustración 1.1 se puede observar un ejemplo hipotético de una RB, donde existen nodos que representan dos enfermedades que son la tifoidea y la gripe, además de la información de la dependen entre los nodos que muestran los síntomas de cada una de ellas [17].



**Ilustración 2. 1. Ejemplo de una Red Bayesiana**

### 2.1.2. Aprendizaje de Redes Bayesianas.

El obtener una red Bayesiana a partir de datos, es un proceso de aprendizaje que consta de dos partes fundamentales, determinar la parte estructural de la red es decir los enlaces entre los nodos que representan las variables y la parte paramétrica que son las tablas de probabilidades asociadas a cada nodo [11].

Típicamente, este problema se divide en dos aspectos:

- Aprendizaje estructural: obtener la estructura de la RB, es decir, las relaciones de dependencia e independencia condicional entre las variables involucradas.
- Aprendizaje paramétrico: consiste en encontrar los parámetros asociados a una estructura dada de una RB.

#### **2.1.2.1. Aprendizaje paramétrico de redes bayesianas.**

El aprendizaje paramétrico reside en encontrar los parámetros asociados a una estructura dada de una RB. Dichos parámetros consisten en las probabilidades a priori de los nodos raíz y las probabilidades condicionales de las demás variables, dados sus padres [11].

Si se conocen datos con todas las variables, es fácil obtener las probabilidades requeridas. Las probabilidades previas corresponden a las marginales de los nodos raíz, y las condicionales se obtienen de las conjuntas de cada nodo con su(s) padre(s) [11] [17].

#### **2.1.2.2. Aprendizaje Estructural de redes bayesianas.**

El aprendizaje estructural de RB reside en encontrar las relaciones de dependencia entre las variables, de tal manera que se pueda establecer la estructura de la red bayesiana. Según el tipo de estructura, podemos dividir los métodos de aprendizaje estructural en: aprendizaje de árboles, aprendizaje de poli árboles y aprendizaje de redes múltiplemente conexas [17].

##### **2.1.2.2.1. Algoritmo para árboles.**

El algoritmo para árboles se basa en el algoritmo desarrollado por Chow y Liu [18] para aproximar una distribución de probabilidad por un producto de probabilidades de segundo orden (árbol) [17].

##### **2.1.2.2.2. Algoritmo para poli árboles.**

El algoritmo para poli árboles [19] el cual es una extensión del algoritmo de Chow y Liu [18] para poli árboles, este se basa en probar las relaciones de dependencia entre todas las tripletas de variables en el esqueleto.

##### **2.1.2.2.3. Algoritmo para redes múltiplemente conexas.**

Existen dos clases de métodos para el aprendizaje genérico de RB, que incluyen redes múltiplemente conexas.

1. **Métodos basados en medidas de ajuste y búsqueda:** se generan diferentes estructuras y se evalúan respecto a los datos utilizando alguna medida de ajuste. [11].
2. **Métodos basados en pruebas de independencia o basados en restricciones:** Estos métodos encuentran un único modelo basado en la información categórica de la independencia condicional entre las variables [11].

### **2.1.3. Metaheurísticas Poblacionales.**

Conforme [4] [7] las metaheurísticas basadas en población son aquellas que emplean un conjunto de soluciones en cada iteración del algoritmo. Estas metaheurísticas poblacionales proporcionan de forma intrínseca un mecanismo de exploración paralelo del espacio de soluciones, y su eficacia depende en gran medida de cómo se manipule dicha población.

Además en [7] [20] se hace referencia que dentro de esta clasificación se destacan los Algoritmos Evolutivos (AEs) y los algoritmos basados en inteligencia colectiva. Entre los más destacados de los algoritmos Evolutivos se encuentran los Algoritmos Genéticos (AGs), Búsqueda Dispersa, Evolución Diferencial y Algoritmos Basados en estimación de Distribuciones (EDA), mientras que en los algoritmos basados en Inteligencia colectiva se encuentran el Sistemas de Partículas, Optimización basada en colonia de hormigas y la optimización basada en mallas variables.

### **2.1.4. Optimización Basada en Mallas Variables (VMO).**

La Optimización Basada en Mallas Variables (Variable Mesh Optimization; VMO) [3] [21] 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

#### **2.1.4.1. 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 variable 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 [3]:

Fase 1: Generación de la población inicial

Fase 2: Generación de nodos hacia los mejores vecinos

Fase 3: Generación de nodos hacia la mejor solución

Fase 4: Generación de nodos utilizando los nodos fronteras de la malla.

El modelo se rige por los siguientes parámetros:

- Cantidad de nodos de la malla inicial ( $N_i$ ).
- Cantidad máxima de nodos de la malla en cada ciclo ( $N$ ), donde  $3 \cdot N_i \leq N$ .
- Tamaño de la vecindad ( $k$ ).
- Condición de parada ( $M$ )

A continuación se muestra una descripción más detallada de cada uno de estos pasos:

- **Generación aleatoria de la malla inicial.**

Para cada uno de los nodos de la malla inicial se valoriza aleatoriamente cada dimensión con un valor real entre el intervalo definido en cada caso.

- **Generación de nodos en dirección a los extremos locales.**

Para calcular los vecinos más cercanos de cada nodo de la malla se utiliza como función de distancia la euclidiana, definida por:

$$D_{euclidiana}(n_1, n_2) = \sqrt{\sum_{i=1}^m (n_1(x_i) - n_2(x_i))^2} \quad (2.1)$$

El factor ( $r$ ), que determina la cercanía del nuevo nodo al nodo actual o al extremo local, se calcula usando la ecuación siguiente:

$$r = \frac{1}{1 + |FO(n) - FO(ne)|} \quad (2.2)$$

esta forma de calcular  $r$  se puede utilizar lo mismo para minimizar que para maximizar una función objetivo, ya que solo mide la razón de separabilidad entre dos valores, no interesa el caso de estudio. La función  $f$  para la generación de nuevos nodos a partir de cada nodo de la malla inicial, que no sea extremo local y el mejor vecino, se define por la ecuación:

$$n_f^*(x_j) \begin{cases} \text{ave}(n_i(x_j), nl_i(x_j)), & (|\text{ave}(n_i(x_j), nl_i(x_j)) - nl_i(x_j)| > D) \text{ and } U[0,1] \leq r_i \\ nl_i(x_j) + U[-D, D], & \text{si } |\text{ave}(n_i(x_j), nl_i(x_j)) - nl_i(x_j)| \leq D \\ U[n_i(x_j), (n_i(x_j)), nl_i(x_j))] & \text{en otro caso} \end{cases} \quad (2.3)$$

Donde  $\text{ave}(n_i(x_j), nl_i(x_j))$  representa el valor medio entre el nodo actual y el extremo local para la  $i$ -ésima dimensión y se calcula como:

$$ave(n(x_j), nl(x_j)) = \frac{n(x_j) + nl(x_j)}{2} \quad (2.4)$$

- **Generación de nodos en dirección al extremo global.**

En este paso de la generación se crean nuevos nodos a partir de cada nodo de la malla inicial en dirección al extremo global ( $ng$ ). El nuevo nodo ( $n_g^*$ ) es generado usando la ecuación 2.5:

$$n_g^*(x_j) = g(n_i(x_j), ng(x_j), r_i) \quad (2.5)$$

La función  $g$  se representa de la siguiente manera:

$$n_g^*(x_j) = \begin{cases} ave[n_i(x_j), ng_i(x_j)], & \text{si } U[0,1] \leq r_i \\ U[ave[n_i(x_j), ng_i(x_j)], ng_i(x_j)], & \text{en otro caso} \end{cases}$$

(2.6)

- **Generación de nodos a partir de los más externos de la malla.**

En este paso se completa la cantidad total de nodos que debe tener la malla, a partir de los nodos fronteras. Para detectar este tipo de nodos se utiliza, para este caso de estudio, el valor de la norma de cada uno, definida por la ecuación:

$$\|n\| = \sqrt[2]{\sum_{i=1}^{nm} (n(x_j))^2} \quad (2.7)$$

Los nodos de mayor norma son los que están situados en el contorno (puntos más externos) de la malla inicial y los de menor norma se consideran los nodos que más cerca se encuentran del origen  $0_m$  (puntos más internos). La función  $h$  permite generar nuevos

nodos en dirección de las fronteras definidas para este caso de estudio, mediante las expresiones:

Para los nodos más externos:

$$n_h(x_j)^* = \begin{cases} nv(x_j) + w, & \text{si } nv(x_j) > 0 \\ nv(x_j) - w, & \text{si } nv(x_j) < 0 \end{cases} \quad (2.8)$$

Para los nodos más internos:

$$n_h(x_j)^* = \begin{cases} |nu(i) - w|, & \text{si } nu(x_j) > 0 \\ |nu(i) + w|, & \text{si } nu(x_j) \leq 0 \end{cases} \quad (2.9)$$

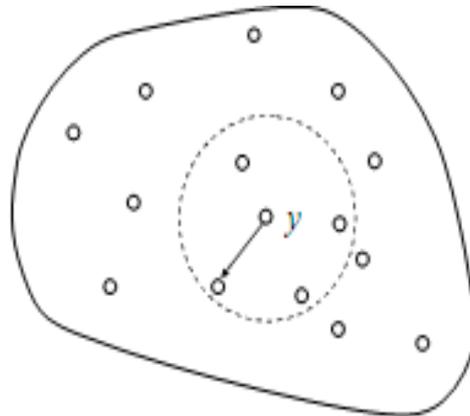
donde el desplazamiento  $w$  se calcula como:

$$w = (w_0 - w_f) \cdot \frac{M - j}{M} + w_f \quad (2.10)$$

donde el parámetro  $M$  y la variable  $j$ , están estrechamente relacionados y provocan las variaciones en el valor de  $w$ ; el primero representa el número total de iteraciones del algoritmo o un número máximo de evaluaciones de la función objetivo (criterio más utilizado). Por su parte,  $j$  denota el valor actual de uno de los criterios mencionados anteriormente. La variable  $w_0$  representa el desplazamiento inicial y  $w_f$  el valor final de este (si  $w_0 > w_f$  efecto decreciente). Para obtener desplazamientos decrecientes relacionados con las amplitudes de cada función, se desarrolló una propuesta adaptativa, donde  $w_0 = am_{DOM}/10$  y  $w_f = am_{DOM}/100$ .

### 2.1.5. Búsqueda Local.

Los algoritmos de Búsqueda Local, definen un proceso de transformación que parte de una solución inicial generada de forma aleatoria o mediante algún otro algoritmo la cual constituye la solución actual. A esta solución se le aplican dicho método para obtener un conjunto de soluciones en un entorno cercano. O sea mediante un proceso iterativo dado una solución actual busca una mejor solución en su vecindario (Ilustración 2).



**Ilustración 2. 2. Espacio de búsqueda**

#### **2.1.6. Metaheurísticas poblaciones con búsqueda local.**

- **Algoritmo híbrido con múltiples colonias (ACOR):** Se centra en algoritmos de colonias de hormigas para resolver problemas de optimización continua. En concreto se propone una extensión de este tipo de algoritmo hacia un método multi-colonia para incrementar la capacidad de exploración. Así mismo, para mejorar las soluciones de generadas por las hormigas artificiales incorpora métodos de BL como el Simplex y el Solis West.
- **Algoritmo Basado en Encadenamiento de Búsquedas Locales para Problemas de Optimización Continua:** AM diseñado para métodos de BL que adaptan los parámetros para guiar la búsqueda y obtener así soluciones más efectivas. Este proceso de adaptación implica una mayor intensidad de aplicación de la Búsqueda Local lo que dificulta su uso dentro de los AM. El algoritmo propone aplicar BL a una solución mediante el encadenamiento de consecutivas aplicaciones de BL
- **AMCR-BLC<sup>2</sup> P:** Algoritmo que utiliza procedimientos de Búsqueda Local basados en el operador de cruce BLC. Dado que este operador produce hijos

alrededor de sus padres puede considerarse como mecanismo para la generación de nuevas soluciones dentro de un método de BL.

### 2.1.7. Búsqueda en Entorno Variable en el espacio de Grafos Dirigidos Acíclicos.

En la tesis doctoral de José Miguel Puerta, se adaptó la metaheurística VNS al problema del aprendizaje de redes bayesianas en el espacio de grafos dirigidos acíclicos, desarrollando el algoritmo VNSST.

Diseñando una transformación específica para este esquema basada en la semántica que tiene una red bayesiana. Concluyendo que mediante la experimentación realizada se puede concluir que este método es muy eficaz y que se mejoran de forma muy significativa los resultados ofrecidos por un esquema de reinicios aleatorios.

---

#### Algoritmo VNSST

---

1. *Inicialización: Fijamos la estructura de vecindades definidas por la ecuación 2.3. Fijamos  $k_{max}$ ,  $k_{min}$  y  $k_{step}$ . Elegimos una solución inicial dada  $G$ . Fijamos número máximo de iteraciones,  $It$ .*
  2. *Ejecutamos el algoritmo HCST con esta solución. Fijar  $G_{max}$ .*
  3.  $n = 1$ .
  4. *Repetir hasta alcanzar la condición de parada. ( $n > It$  ó dos iteraciones sin mejoras).*
    - (a)  $k = k_{min}$  y  $n = n + 1$ .
    - (b) *Repetir hasta que  $k = k_{max}$* 
      - i. *Generar de forma aleatoria una  $G'$  de vecindad  $k$  ( $G' \in \mathcal{N}_k(G_{max})$ ).*
      - ii. *Aplicar el algoritmo HCST con  $G'$  como punto de inicio, dando lugar a otro óptimo local  $G''$ .*
      - iii. *Si  $f(G'' : D)$  es mejor que  $f(G_{max} : D)$ , entonces;  $G_{max} = G''$  y  $k = k_{min}$ ; en otro caso  $k = k + k_{step}$ .*
- 

### Ilustración 2. 3 Pseudocódigo del algoritmo VNSST

#### 2.1.8. Métrica k2.

La métrica K2 [10] es una medida de evaluación bien conocido para el aprendizaje de redes bayesianas a partir de datos, que utiliza una métrica de puntuación bayesiana para medir la probabilidad conjunta de un BN. La métrica de puntuación se conoce como métrica K2 como se utiliza en el algoritmo K2. La expresión inicial de la métrica K2 es:

$$P(G, D) = P(G) \prod_{l=1}^N \prod_{j=1}^{q_l} \frac{(r_l-1)!}{(N_{lj}+r_l-1)!} \prod_{k=1}^{r_l} N_{ljk}! \quad (2.11)$$

donde D es un conjunto de entrenamiento dado, G es una posible estructura de la red,  $r_l$  es el número de valores posibles de la variable  $X_l$ ,  $q_l$  es el número de posibles configuraciones (instanciaciones) para las variables en  $(X_l)$ , y  $N_{ljk}$  es el número de los casos en D, donde  $X_l$  tiene su valor  $k$ -ésimo y  $(X_l)$  se instancia a su valor  $j$ .

### 2.1.9. ACO-B.

El algoritmo ACO-B [10] es un enfoque de puntuación y búsqueda basada en la optimización de colonia de hormigas para el aprendizaje de las redes bayesianas, cuya idea principal es utilizar la métrica K2 para evaluar una estructura RB para buscar el máximo global en un espacio solución factible.

Sea  $a$  el número de hormigas en una colonia de hormigas,  $I_j(t)$  la intensidad de feromonas asociado con el arco dirigido  $a_{ij}$  en el momento  $t$ , y la intensidad de feromonas inicial de cada arco dirigido ser un valor constante  $C$ , es decir,  $t_{ij}(0) = C$ . una feromona es una sustancia química liberada en el medio ambiente que puede reflejar la comunicación entre las hormigas. En la construcción de una solución, cada  $k$  hormiga ( $k = 1, 2, \dots, A$ ) se inicia desde el  $G_0$  gráfico de vacío (arcos-less DAG) y procede mediante la adición de un arco a la vez.

**Algorithm: ACO-B**

1. **Initialization:**  
Initialize  $a$ ,  $NC$ ,  $G(0)$ ,  $\rho$ ,  $\varphi$ ,  $q_0$ ,  $G^+ = G(0)$ ,  $\tau_0$  and  $l_{step}$
2. **Loop:**  
For  $l=1$  to  $NC$  do:  
1) For  $k=1$  to  $a$  do:  
     $G_k = AntConstructGraph()$ ;  
    If  $(l \bmod l_{step}=0)$  then  $G_k = Optimization(G_k)$ ;  
2)  $G_{(l)}^+ = argmax_{k=1 \dots a} f(G_k : D)$ ;  
3) If  $(f(G_{(l)}^+ : D) \geq f(G^+ : D))$  then  $G^+ = G_{(l)}^+$  ;  
4) Perform global pheromone updating by Eq.(8);
3. **Local optimization:**  
1) For  $k=1$  to  $a$  do:  $G_k^o = Optimization(G_k)$   
2)  $G_{(l)}^+ = argmax_{k=1 \dots a} f(G_k^o : D)$ ;  
3) If  $(f(G_{(l)}^+ : D) \geq f(G^+ : D))$  then  $G^+ = G_{(l)}^+$  ;
4. **Return**  $G^+$ ;

**Ilustración 2. 4. Pseudocódigo del algoritmo ACO-B.**

### 2.1.10. Test de Friedman

El test de Friedman [7] constituye un análisis de varianza de segunda vía para variables apareadas o de carácter no paramétrico. Haya el orden de los resultados observados por 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:

$$x_F^2 = \frac{12N}{k \cdot (k+1)} \left[ \sum jR_j^2 - \frac{k \cdot (k+1)^2}{4} \right], \quad (2.12)$$

se distribuye aproximadamente acorde a  $x^2$  con  $k - 1$  grados de libertad, siendo  $R_j = \frac{1}{N} \sum_i r_i^j$ , 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  $k > 5$ .

## **CAPÍTULO II**

# **METODOLOGÍA DE LA INVESTIGACIÓN**

### **3.1. Localización.**

Este proyecto se encamina dentro del campo de la investigación de la unidad de Titulación Especial de la Universidad Técnica Estatal de Quevedo, que se encuentra ubicada en el cantón Quevedo, de la provincia de los Ríos.

### **3.2. Tipo de Investigación.**

#### **3.2.1. Exploratoria**

La investigación exploratoria permitirá introducir la propuesta de aplicar búsqueda local al algoritmo basado en mallas variables que fue desarrollado en [9] para el problema discreto de entrenamiento de redes bayesianas.

### **3.3. Métodos de Investigación.**

#### **3.3.1. Analítico**

El método analítico [22] permitió interpretar la situación desde la perspectiva analítico – crítico, para así obtener un mejor conocimiento del problema de estudio y así cumplir con el fin propuesto en el presente proyecto de investigación.

### **3.4. Fuentes de recopilación de información.**

La información necesaria para el desarrollo de la investigación se la obtuvo de fuentes secundarias, dado que se recopiló la información a partir de artículos y tesis relacionadas sobre el problema de estudio y los diferentes algoritmos.

### **3.5. Diseño de Investigación.**

#### **3.5.1. Diseño Cuasi – Experimental.**

El diseño de investigación que se utilizará es el diseño cuasi-experimental, para evaluar el efecto que tendrá aplicar algoritmos de búsqueda local al algoritmo basado en mallas variables.

### **3.5.2. Pasos de un cuasi experimento.**

Este diseño de investigación será realizado mediante un estudio comparativo entre la aplicación de búsqueda local al algoritmo VMO con otros algoritmos mencionados en el estado del arte.

Esquema del diseño O1 X O2

Dónde:

X = Algoritmo basado en mallas variables con búsqueda local.

O1 = Medición antes del experimento (Resultados obtenidos del VMO con algoritmo de búsqueda local).

O2 = Medición después del experimento (Resultados obtenidos de los algoritmos ACOB, ACOB1, ACOB2, ACOB3).

### **3.6. Instrumentos de Investigación.**

El instrumento de investigación que se utilizará para el desarrollo del proyecto fue el análisis de documentos, que permitirá obtener información relacionada con los diferentes algoritmos a utilizar en la investigación.

### **3.7. Tratamientos de los datos.**

Para el tratamiento de los datos se utilizó la herramienta ofimática Microsoft Excel la que permitirá generar el archivo CSV (Valores Separados por Comas) con los resultados obtenidos con la ejecución del algoritmo VMO con cada uno de los algoritmos de búsqueda local, para luego realizar el Test de Friedman e Iman – Davenport para comprobar si existe o no diferencia significativa entre ellos y cuál de los algoritmos de búsqueda local se obtuvieron mejores resultados y proceder a realizar el estudio comparativo con los algoritmos citados en el estado del arte.

### 3.8. Recursos humanos y materiales.

Los recursos humanos y materiales que se utilizaron en el desarrollo del presente proyecto de investigación son los siguientes:

#### 3.8.1. Hardware.

Tabla 1. Materiales de hardware.

CANTIDAD	MATERIAL	DESCRIPCIÓN
		HP Pavilion g4-20551a.
		<b>Características:</b>
1	Computador	<ul style="list-style-type: none"><li>• Intel Core TM i5-2450M.</li><li>• 8 Gb. RAM.</li><li>• 750 Gb Disco Duro.</li><li>• Cd Writer.</li></ul>
1	Impresora	EPSON L355

#### 3.8.2. Software.

Tabla 2. Materiales de software.

CANTIDAD	MATERIAL	DESCRIPCIÓN
1	Sistema Operativo	Microsoft Windows: versión Windows 7.
		Microsoft Office 2010:
1	Procesador de Texto	<ul style="list-style-type: none"><li>• Word.</li><li>• Power Point.</li></ul>

### 3.8.3. Suministros.

Tabla 3. Tabla de Suministro.

<b>CANTIDAD</b>	<b>MATERIAL</b>	<b>DESCRIPCIÓN</b>
1	Internet	Plan Internet Banda Ancha Hogar de CNT.
	Materiales	<ul style="list-style-type: none"><li>• 2 Resmas Hojas A4.</li><li>• 7 CD-Rw.</li><li>• 1 Grapadora.</li><li>• 1 caja de grapas.</li><li>• 1 caja de clips.</li><li>• Tinta EPSON 664 – Negro</li></ul>

### 3.8.4. Personal.

Tabla 4. Personal.

<b>PERSONAL</b>	<b>DESCRIPCIÓN</b>
Desarrollador.	Irina Monserrat Pacheco Zamora
Director de Tesis.	Phd. Amilkar Yudier Puris Cáceres.

## **CAPÍTULO IV**

### **RESULTADOS Y DISCUSIÓN**

## 1.4. Resultados.

En esta sección se describe la forma en que se realizó la integración de búsqueda local a la metaheurística basado en mallas variables, utilizando la métrica  $k_2$ . Los valores se obtuvieron mediante la implementación del algoritmo VMO con cada algoritmo de búsqueda local que viene integrado en el software Elvira. Para luego realizar un estudio comparativo entre el algoritmo VMO con búsqueda local que mejor resultados obtuvo con los algoritmos citados en el estado del arte.

En la investigación “Aprendizaje estructural de redes bayesianas utilizando la metaheurística optimización basada en mallas variables (VMO)” [23] se realizó la propuesta de encontrar una forma de aplicar el VMO al problema discreto de entrenamiento estructural de redes bayesianas, ya que no se había presentado ninguna solución que obtenga los mejores resultados en para cualquier base de conocimiento; para la ejecución de la investigación y para proceder a realizar un estudio comparativo entre los resultados que se obtuvieron con el algoritmo VMO con algunos clasificadores bayesianos como ByNet, BayesChaid, BayesPSO, RB K2, RB TAN, CBN, Naïve Bayes y Naïve Bayes aumentado a árbol (TAN), se utilizaron 14 bases del conocimiento internacionales con diferentes características que se encuentran en la UCIML.

En la sección de resultados del proyecto anteriormente mencionado, se describe con más detalles la forma en que se ajustaron algunos de los elementos la metaheurística poblacional basada en mallas variables, desde cómo se realiza la definición del nodo, cálculo de distancia entre nodos, generación de nuevas redes a partir de un par y la selección de las mejores redes hasta la aplicación del algoritmo VMO al aprendizaje estructural de redes bayesianas.

Además que para la experimentación, se probaron diferentes variantes el algoritmo con los diferentes valores de parámetros (tamaño de la malla inicial  $p = (12, 24, 48)$ , tamaño de las vecindades  $k = (3, 5, 7)$  y cantidad de iteraciones = 1000), donde cada variante fue ejecutada de manera independiente para cada base de conocimiento.

Para los resultados de cada variante se aplicaron test estadísticos para detectar si existe o no diferencias significativas. En todo el proceso de pruebas se logró definir a la variante VMO con cantidad de nodos de malla inicial de 24, tamaño de la vecindad de 3 como la

que obtuvieron mejores resultados significativamente mayores que las demás variantes estudiadas.

#### 4.1.1. Estudio experimental de la aplicación de búsqueda local al algoritmo VMO.

Para la realización de la presente investigación se utilizan los parámetros que se mencionaron anteriormente, como son:

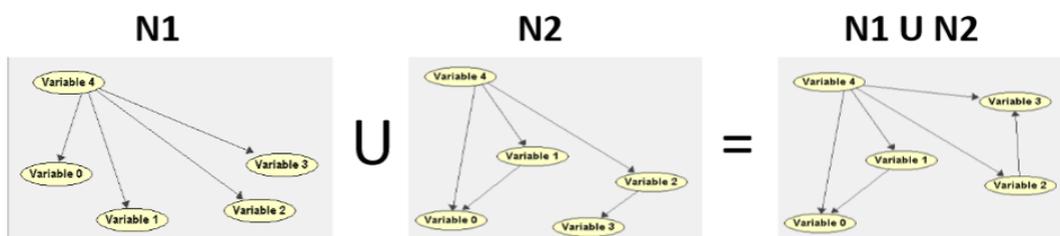
Cantidad de nodos de malla inicial = 24.

Tamaño de vecindad = 3.

Numero de iteraciones =1000.

Además se utilizara la propuesta de nuevas soluciones a partir del operador unión entre conjuntos, el funcionamiento de este operador se puede observar en la *ilustración 3*. Así como la contracción de la malla utilizando el explorador elitista que se encarga de ordenar de mayor a menor los nodos basándose en una métrica de calidad y se selecciona los M mejores nodos dependiendo del parámetro de la malla inicial.

**Ilustración 1. Ejemplo del operador unión**



#### 4.1.2. Integración del algoritmo VMO con búsqueda local en el software Elvira.

**Paso 1 [Generación malla inicial]:** se genera una población inicial de M nodos (redes) de manera aleatoria mediante la llamada al método run() de la clase principal VMO.java, después de la generación de cada nodo se le calcula la métrica mediante el método CalculatePrecision(Bnet Net,int k, int type), al cual se envían como parámetros una red bayesiana, un indicador k y el tipo de generación para saber en qué paso fue generada la red. Este cálculo se almacena en una variable para futuros procedimientos.

**Paso 2 [Generación por extremo local - LocalExtreme(TamNeigh)]:** en este caso se calcula una matriz de distancia que contiene valores de las diferencias entre los nodos, este proceso lo realiza el método HammingMatrix() ubicado en la clase VMO\_MESH.java, después de esto seleccionamos la red más cercana de mejor calidad y se obtiene una nueva red con el método BestNetUnion(Vector vector, int k,int type) al cual le envío lo parámetro de 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()]:** en este paso se consideró la idea original la cual obtiene la red de mejor calidad y se generan nuevas redes a partir de esa.

**Paso 4 [Generación por extremo Frontera - BorderNode()]:** Se se seleccionan los nodos más distantes, es decir el mejor y el peor en cuanto a calidad y se le cambia la dirección de algún arco aleatoriamente.

**Paso 5 [Seleccionar la malla inicial para la siguiente iteración - setNetSolVMO()]:** en este paso seleccionamos las M mejores redes comparándolas mediante la calidad obtenida por su estructura.

**Paso 6 [Aplicación de búsqueda local]:** para el proceso de aplicación de búsqueda local se utilizan los métodos que vienen incluidos en el Elvira, el cual tiene implementados los algoritmos que hemos utilizado dentro de esta investigación como los son ThVNSSTPC.java, ThVNSST.java, ThVNSST2.java, enviándole como parámetro la red bayesiana a la cual se le aplicara la búsqueda local, los datos con los cuales se realizó el aprendizaje, la métrica, un grafo vacío y el fitness actual de la red.

Cada uno de estos pasos se los realiza hasta que llegue a una cantidad de iteraciones de 1000. Además que el paso 6 se lo realizará dependiendo de la intensidad de la búsqueda local.

### **4.1.3. Análisis de los resultados.**

Para la experimentación se utilizó la base de conocimiento *alarm* diseñada para proporcionar un sistema de mensajes de alarma a la monitorización del paciente, la cual consta de 6000 registros y 37 variables.

Para evaluar la incorporación de la búsqueda local al algoritmo VMO, se realizó la subdivisión de la base de conocimiento *alarm*, de tal forma como se lo realiza en [10] para así realizar un estudio comparativo con los algoritmos del estado del arte.

La experimentación se realizó de la siguiente manera:

1. Obtener la hibridación del algoritmo VMO con los diferentes algoritmos de búsqueda local que mejores resultados obtuvieron.
2. Comparar el algoritmo VMO con búsqueda local seleccionada anteriormente con los diferentes algoritmos del estado del arte.

La plataforma experimental que se utilizó para ejecutar unas 50 veces cada implementación del algoritmo con búsqueda local para cada tamaño de la base de conocimiento, fue una PC con core i7, 3,5 GHz CPU, memoria 16GB, y Windows 7.

#### **4.1.3.1. Estrategias de Búsqueda local.**

Se aplicaron tres algoritmos de búsqueda local, a diferentes tamaños de muestra de la base de conocimiento *alarm*. Los tres algoritmos fueron: VNSSTPC, VNSST, VNSST2 con la métrica k2. Así mismo se definió que se realizarán pruebas con una intensidad de 150, 250, 350.

La primera etapa de pruebas se la realizó con una intensidad de cada 150 iteraciones en la cual se obtuvieron los siguientes resultados:

Tabla 5. Resultados con una intensidad de 150 iteraciones.

Sample capacity	VMO_SEB	VMO-VNSST	VMO-VNSSTDOS	VMO-VNSSTPC
1000	-7846.90	-5698.77	-4689.26	-5783.50
2000	-14868.62	-9238.05	-93475.67	-98764.04
3000	-22230.67	-14879.58	-13489.45	-13532.02
4000	-29211.15	-17989.98	-17945.12	-19753.34
5000	-38063.01	-22986.62	-21222.97	-23910.11
6000	-42828.97	-25993.34	-27912.01	-26983.03

La segunda etapa de pruebas se la realizó con una intensidad de cada 250 iteraciones en la cual se obtuvieron los siguientes resultados:

Tabla 6. Resultados con una intensidad de 250 iteraciones

Sample capacity	VMO_SEB	VMO-VNSST	VMO-VNSSTDOS	VMO-VNSSTPC
1000	-7846.90	-4407.41	-4432.55	-4381.52
2000	-14868.62	-8437.35	-8468.02	-8392.21
3000	-22230.67	-12575.58	-12583.59	-12611.77
4000	-29211.15	-16689.26	-16683.64	-16650.77
5000	-38063.01	-20755.01	-20762.70	-20753.05
6000	-42828.97	-24887.89	-24857.89	-24798.63

La tercera etapa de pruebas se la realizó con una intensidad de cada 350 iteraciones en la cual se obtuvieron los siguientes resultados:

Tabla 7. Resultados con una intensidad de 350 iteraciones

Sample capacity	VMO_SEB	VMO-VNSST	VMO-VNSSTDOS	VMO-VNSSTPC
1000	-7846.90	-5867.58	-5345.89	-5684.01
2000	-14868.62	-9635.78	-8923.68	-8457.32
3000	-22230.67	-13986.89	-13579.03	-13986.12
4000	-29211.15	-17983.43	-17934.34	-16823.06
5000	-38063.01	-24723.09	-24567.56	-24679.07
6000	-42828.97	-25634.76	-46676.89	-24856.98

Como se puede observar en las tablas 5, 6 y 7 con los resultados que se obtuvieron implementando búsqueda local al algoritmo VMO, en cada una de las etapas de pruebas se utilizaron los parámetros anteriormente mencionados, que son la cantidad de nodos de malla inicial de 24, el tamaño de vecindad de 3, número de iteraciones de 1000 y una intensidad de 150, 250 y 350 respectivamente. Donde se obtuvieron mejores resultados con una intensidad de 250.

La forma en que se realizó la integración de búsqueda local al VMO, con cada una de las intensidad (150, 250, 350) fue de la siguiente manera:

En este caso con una de intensidad de 250 y para cada uno de los algoritmos VNSSTPC, VNSST, y VNSST2. desde la primera iteración hasta llegar a la iteración 249 se realizaron los primeros pasos del algoritmo VMO (ver la sección 4.1.2 del presente capítulo), una vez que se llega a la iteración 250 se aplican los cuatros primeros pasos del VMO más el paso de ejecutar el algoritmo de búsqueda local, el cual escoge el mejor nodo, una vez terminado esta fase, se vuelve a ejecutar los primero 4 pasos hasta llegar a la iteración 500 donde se va a volver aplicar búsqueda local y así sucesivamente hasta llegar a la iteración 750 y la iteración 999 escogiendo siempre el mejor nodo.

Una vez que se definió la intensidad de la búsqueda local, se procedió a aplicarle el test de Friedman e Iman Davenport, donde se obtuvieron los resultados que se muestran en la tabla 8. Utilizando 4 grados de libertad para la distribución de Chi Cuadrado al aplicar el test de Fridman así como 4 y 20 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 medias y se concluye que existe diferencias significativas entre los algoritmos comparados.

Tabla 8. Test De Friedman E Iman Davenport VMO-Búsqueda Local

<b>TEST</b>	<b>VALOR DE TEST</b>	<b>VALOR DE DISTRIBUCIÓN</b>	<b>VALOR P</b>	<b>HIPÓTESIS</b>
FRIEDMAN	16.3999	9.499	0.0025	RECHAZADA
IMAN DAVENPORT	10.78947	2.87	7.942735	RECHAZADA

En este caso se selecciona el algoritmo de control VNSSTPC, ya que es significativamente mejor que los demás algoritmos búsqueda local comparados y alcanza mayor valor medio.

#### 4.1.4. Análisis comparativo del algoritmo VMO con búsqueda local con el estado del arte.

Una vez seleccionada la hibridación del algoritmo VMO con el algoritmo búsqueda local, con el que se obtuvieron mejores resultados, se procedió a realizar el estudio comparativo con los algoritmos del estado del arte. Dónde:

- **ACO – B:** El algoritmo ACO-B es un enfoque de puntuación y búsqueda basada en la optimización de colonia de hormigas para el aprendizaje de las redes bayesianas, utilizando la métrica k2.
- **ACO-B1:** sólo utilizando espacios de búsqueda de variables.
- **ACO-B2:** sólo mediante la nueva función heurística.
- **ACO-B3:** sólo mediante la estrategia de recocido simulado.

En la tabla 9. Se puede observar los resultados del algoritmo de Control seleccionado VMO-VNSSTPC anteriormente con cada uno de los algoritmos.

Tabla 9. Resultados de la comparación entre la hibridación de Búsqueda con el Algoritmo VMO y los algoritmos citados en el estado del arte.

Sample capacity	VMO-VNSSTPC	ACO-B	ACO-B1	ACO-B2	ACO-B3
1000	-4381.52	-5024.14	-5023.66	-5024.16	-5024.56
2000	-8392.21	-9717.64	-9717.79	-9717.57	-9718.25
3000	-12611.77	-14402.01	-14401.54	-14401.88	-14402.10
4000	-16650.77	-19099.64	-19098.76	-19099.26	-19100.18
5000	-20753.05	-23782.17	-23782.12	-23782.72	-23782.55
6000	-24798.63	-28347.17	-28347.50	-28347.67	-28347.50

Se procedió a aplicarle el test de Friedman e Iman Davenport, donde se obtuvieron los resultados que se muestran en la tabla 10. Utilizando el valor de la distribución de Chi Cuadrado con 4 grados de libertad para el test de Friedman, así como el valor de la distribución de Fisher con 4 y 20 grados de libertad para la aplicación del 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 y se concluye que existe diferencias significativas entre los algoritmos comparados.

Tabla 10. Test de Friedman e Iman Davenport VMO+VNSSTPC - estado del arte.

<b>TEST</b>	<b>VALOR DE TEST</b>	<b>VALOR DE DISTRIBUCIÓN</b>	<b>VALOR P</b>	<b>HIPÓTESIS</b>
FRIEDMAN	17.16666	9.499	0.00179	RECHAZADA
IMAN DAVENPORT	12.56097	2.87	2.85442	RECHAZADA

En este caso se selecciona el algoritmo de control VNSSTPC, ya que es significativamente mejor que los demás algoritmos comparados y alcanza mayor valor medio. Además se pudo deducir que aplicar búsqueda local a algoritmo VMO se obtuvieron mejores resultados en comparación con los algoritmos del estado del arte.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

## 5.1. Conclusiones.

Con la culminación de la presente investigación sobre la incorporación de búsqueda local al algoritmo basado en mallas variables para el aprendizaje estructural de redes bayesianas se logró llegar a las siguientes conclusiones:

- Se logró hibridar el algoritmo VMO con los algoritmos de búsqueda en entorno variables con los parámetros ya establecidos de cantidad de malla inicial de 14, tamaño de vecindad de 3, números de iteraciones de 1000 y utilizando la métrica  $k_2$ .
- Se realizaron pruebas con una intensidad de 150, 250 y 350 para determinar cada cuando aplicar la búsqueda local al algoritmo VMO, donde se obtuvieron mejores resultados aplicando una intensidad de 250 en todos los algoritmos de VNS.
- Se diseñaron distintas variantes de hibridación del algoritmo VMO con cada uno de los algoritmos VNSST, VNSST2 y VNSSTPC de búsqueda local utilizando la métrica  $k_2$ . Donde se logró definir que la mejor hibridación fue con el algoritmo VNSSTPC.
- Se realizó un estudio comparativo entre la variante seleccionada VMO-VNSSTPC con los diferentes algoritmos citados en el estado del arte (ACO-B, ACOB-1, ACOB-2, ACOB-3), obteniendo resultados significativamente mejores con respecto a los algoritmos estudiados.

## **5.2. Recomendaciones.**

Al finalizar la investigación surgen las siguientes recomendaciones:

- Implementar un estudio donde se incluyan más algoritmos de búsqueda local al algoritmo basado en mallas variables, además que se realice un estudio comparativo con la actual implementación.
- Estudiar y analizar nuevos valores de intensidad de búsqueda local al algoritmo basado en mallas variables.
- Se recomienda ampliar el presente estudio con otros algoritmos, para comprobar la efectividad de la implementación de búsqueda local al algoritmo VMO.

## **CAPÍTULO VI**

## **BIBLIOGRAFÍA**

## 6.1. Bibliografía.

- [1] Pearl Judea, Probabilistic reasoning in intelligent systems. Morgan Kaufmann, San Mateo, 1988.
- [2] A. Puris, R. Bello, D. Molina, F. Herrera, Variable mesh optimization for continuous optimization problems, 2012.
- [3] E. Díaz, A. Puris y R. Bello, Comparación del desempeño entre los algoritmos VMO, DE y ODE, 2012.
- [4] Duarte Muñoz Abraham, Metaheurísticas, Dykinson, 2007.
- [5] Navarro Ricardo, Puris Amilkar, Bello Rafael y Herrera Francisco, «Estudio del desempeño de la optimización basada en mallas variables en problemas con óptimos en las fronteras del espacio búsqueda,» *RCCI Vol. 3*, 2009.
- [6] Glover F. y Kochenberger G., Handbook of Metaheuristics, Kluwer Academic Publishers, 2003.
- [7] Martínez Carlos, Desarrollo de un Algoritmo Memético basado en VMO., Santa Clara, 2010, p. 7.
- [8] «Bayesian Network Repository,» [En línea]. Available: <http://www.bnlearn.com/bnrepository/>.
- [9] Moreira Zamora Luis Enrique, Aprendizaje estructural de redes bayesianas utilizando la meta-heurística optimización de mallas variables (VMO), Quevedo, Los Ríos: UTEQ, 2015.
- [10] Junzhong Ji, Renbing Hu, Hongxun Zhang y Chunnian Lui, «A hybrid method for learning Bayesian networks based on ant colony optimization,» de *Applied Soft Computing*, 2011.
- [11] Lemagne Rafael Chávez, Ambiente de Aprendizaje de Redes Bayesianas basado en

Web, Santa Clara: Universidad Central Marta Abreu de las Villas, 2011.

- [12] Stuart J. Russell y Norvig Peter, *Inteligencia Artificial: Un enfoque Moderno*, Prentice Hall, 1996.
- [13] S. J. Russell y N. Peter, *Artificial Intelligence: A Modern Approach*, Segunda ed., Prentice Hall, 2003, p. 1132.
- [14] Spirtes Peter, Glymour Clark y Scheines Richard, *Causation, Prediction and Search* Springer Verlag, New York.
- [15] Pearl Judea, *Graphical Models, Causality and Intervention*, vol. VIII, *Statistical Science*, 1993.
- [16] B. R., *Bayesian Belief Networks: From Construction to Inference*, Tesis Doctoral de la University Utrecht., 1995.
- [17] Sucar Luis Enrique, *Redes Bayesianas*, Sta. María Tonantzintla,: INAOE, 2010.
- [18] Chow C. y Liu C., *Approximating discrete probability distribution with dependence trees.*, *IEEE Transactions on Information Theory*, 1968.
- [19] Rebane George y Pearl Judea, *The recovery of causal poly- trees from statistical data.*, *Int. J. Approx. Reasoning*, 1988.
- [20] M. Carlos, P. Amilkar y B. Rafael, *Mecanismo de limpieza del espacio de soluciones para fomentar la diversidad de la metaheurística PSO*, Santa Clara, 2012.
- [21] O. Byron, G. Jorge, D. Eduardo y P. Amilkar, «Optimización basada en Mallas Variables: Caso de estudio viajante de Comercio,» de *LACCEI Latin American and Caribbean Conference for Engineering and Technology*, Guayaquil, 2014.
- [22] Torres Bernal César Augusto, *Metodología de la Investigación*, vol. III, Pearson Educación, 2010.

## **CAPÍTULO VII**

### **ANEXOS**