

# Algoritmos meta heurísticos para el aprendizaje de redes bayesianas<sup>1</sup>

Byron Oviedo<sup>1</sup>, Amilkar Puris<sup>2</sup>, Emilio Zhuma<sup>3</sup>

## RESUMEN

**Introducción** Con este artículo se pretende obtener modelos basados en análisis probabilísticos de casos para la ayuda a la toma de decisiones en la educación y aprendizaje de los estudiantes de la UTEQ. Para la obtención del producto final, se ha distribuido el proceso de desarrollo en varias etapas. **Objetivo** Crear un modelo probabilístico para evaluar y diagnosticar a los estudiantes en función de un conjunto de características, las mismas que deberán ser aprendidas de forma automática mediante una generalización del modelo AutoClass permitiendo la existencia de variables ocultas, cada una de ellas afectando a un conjunto distinto de variables observables (respuestas de los alumnos a cuestiones planteadas por un sistema automático de

aprendizaje). **Materiales y Métodos.** Nuestro estudio se llevará a cabo para definir otra forma de aprendizaje estructural basado en la búsqueda de estructuras a través de modelos meta-heurísticas evolutivas. **Resultados** Este modelo permitirá a las autoridades de la UTEQ determinar inconvenientes y contratiempos en el proceso de enseñanza aprendizaje. A la vez los resultados obtenidos permitirán una toma de decisiones inmediata para solucionar los problemas detectados y de esta manera cumplir con la misión institucional de formar profesionales con visión científica y humanista capaces de desarrollar investigaciones, crear tecnologías, mantener y difundir nuestros saberes y culturas ancestrales, para la construcción de soluciones a los problemas de la región y el país. **Conclusiones** se optimización de

1 Artículo original derivado del proyecto de investigación titulado Algoritmos Meta heurísticos para el Aprendizaje de Redes Bayesianas Entidad financiadora: Universidad Técnica Estatal de Quevedo, fecha de realización entre 2015-2016

2 Licenciado en Ciencias de la Computación, Máster en Ciencias de la Computación, Ph.D. en Ciencias Técnicas, Docente Investigador, Universidad Técnica Estatal de Quevedo, apuris@uteq.edu.ec, ORCID: 0000-0002-7288-7451, Scopus ID: 15122856600

3 Ingeniero en Computación, Máster en Conectividad y Redes de Ordenadores, Coordinador de la Carrera de Ingeniería en Telemática, Universidad Técnica Estatal de Quevedo, ezhuma@uteq.edu.ec, ORCID: 0000-0002-3086-1413

Artículo recibido: 1/05/2018; Artículo aprobado: 19/07/2018

Autor para correspondencia, Byron Wladimir Oviedo Bayas, E-mail: boviedo@uteq.edu.ec

malla variable (VMO) metaheurístico al aprendizaje estructural de los clasificadores de redes bayesianas (BVMO).

**PALABRAS CLAVE:** Meta–heurística, AutoClass, Variables Ocultas

## Meta-heuristic algorithms for the learning of bayesian networks

### ABSTRACT

**Introduction** This article aims to obtain models based on probabilistic case analysis to help decision-making in the education and learning of UTEQ students. To obtain the final product, the development process has been distributed in several stages. **Objective** To create a probabilistic model to evaluate and diagnose students based on a set of characteristics, which must be learned automatically through a generalization of the AutoClass model allowing the existence of hidden variables, each of them affecting a set different from observable variables (students' answers to questions raised by an automatic learning system). **Materials and methods.** Our study will be carried out to define another form of structural learning based on the search of structures through evolutionary meta-heuristic models. **Results** This model will allow the authorities of the UTEQ to identify inconveniences and setbacks in the teaching-learning process. At the same time, the results obtained will

allow immediate decision-making to solve the problems detected and thus fulfill the institutional mission of training professionals with a scientific and humanistic vision capable of developing research, creating technologies, maintaining and disseminating our ancestral knowledge and culture, for the construction of solutions to the problems of the region and the country. **Conclusions** were metaheuristic variable mesh optimization (VMO) to structural learning of Bayesian network classifiers (BVMO).

**KEYWORDS:** Meta–heuristics, AutoClass, Hidden Variables

## Algoritmos meta heurísticos para a aprendizagem de redes bayesianas

### RESUMO

**Introdução** O objetivo deste artigo é obter modelos baseados em análise probabilística de casos para auxiliar na tomada de decisões na educação e aprendizagem de alunos da UTEQ. Para obter o produto final, o processo de desenvolvimento foi distribuído em várias etapas. **Objetivo** Para criar um modelo probabilístico para avaliar e diagnosticar alunos com base em um conjunto de características, o mesmo que deve ser aprendido automaticamente por um generalização do modelo AutoClass permitindo a existência de variáveis ocultas, cada afetando um todo diferente das

variáveis observáveis (respostas dos alunos às questões levantadas por um sistema automático de aprendizagem). **Materiais e métodos.** Nosso estudo será realizado para definir outra forma de aprendizagem estrutural baseada na busca de estruturas através de modelos meta-heurísticos evolutivos. **Resultados** Este modelo permitirá que as autoridades da UTEQ identifiquem inconveniências e retrocessos no processo de ensino-aprendizagem. Embora os resultados permitirá tomar decisões imediatas para resolver os problemas identificados e, assim, cumprir a missão institucional de formar

profissionais com visão científica e humanística capaz de desenvolver a investigação, desenvolver tecnologias, manter e disseminar o nosso conhecimento e ancestrais culturas , para a construção de soluções para os problemas da região e do país. **As conclusões** adaptaram a otimização da malha metaheurística variável (VMO) para a aprendizagem estrutural de classificadores de redes bayesianas (BVMO).

**PALAVRAS-CHAVE:** Meta-heurística, AutoClasse, Variáveis Ocultas

---

## INTRODUCCIÓN

La educación se puede ver como un proceso intencional, continuo y sistemático de perfeccionamiento de las personas en cualquiera de sus dimensiones (intelectual, física, estética, profesional, ética, etc.). En su modalidad de proceso, la educación está sujeta a la supervisión constante de todos los procesos de enseñanza y aprendizaje que la componen. La monitorización del proceso educacional, es por tanto un método que analiza constantemente cómo marchan las actividades de enseñanza y aprendizaje en correspondencia con los objetivos propuestos; esto permite garantizar la dirección del proceso hacia una situación deseada, introducir acciones educativas adicionales y obtener la información necesaria y útil para tomar las decisiones que correspondan. Se debe recordar que monitorizar no es sinónimo de evaluar,

aunque sin dudas son procesos que tienen muchos puntos en común.

En la actualidad la deserción y pérdida estudiantil en el sistema de enseñanza superior tiene un índice alto de suspensión o abandono de estudios, ya que por diferentes motivos y circunstancias los estudiantes no logran conseguir la culminación de su carrera estudiantil. Esto es más importante en los cursos a distancia mediante ordenador en los que la interacción y la capacidad de adaptación son muy limitadas en la actualidad. Conocer las razones de la deserción y tratar de construir sistemas de aprendizaje que se adapten mejor a las características de cada alumno son objetivos de indudable interés y que mejorarían el proceso educativo.

En este sentido han sido utilizados distintos métodos de Inteligencia Artificial para el aprendizaje automatizado, con el

objetivo de encontrar relaciones entre variables y cuantificar la importancia de estas para la toma de decisiones. El objetivo de los algoritmos de aprendizaje consiste en determinar un modelo gráfico a partir de un conjunto de datos en bruto u observaciones realizadas sobre el comportamiento del sistema, referencias importantes son (Cooper, 1992) y (Heckerman, 1995).

Existe en la actualidad una gran diversidad de enfoques y métodos de resolución. Dentro del modelo de aprendizaje automatizado más utilizados en el área de la educación se encuentran las redes bayesianas, junto con los árboles de decisión y las redes neuronales artificiales.

En los últimos años los sistemas expertos probabilistas o redes bayesianas han alcanzado un alto grado de desarrollo. Hasta los 80 se había dado por supuesto que la probabilidad requería mucha información y unos cálculos demasiado complejos para poder resolver problemas reales en los que interviniesen un gran número de variables. Sin embargo, esto cambió a partir de una serie de trabajos entre los que destacan los de (Pearl, 1988).

La idea esencial fue la de aprovechar las relaciones de dependencia (y por tanto también las de independencia) existentes entre las variables de un problema antes de especificar y calcular con los valores numéricos de las probabilidades. Estas relaciones se representan a través de modelos gráficos, habitualmente grafos a cíclicos dirigidos (Nielsen, 2009).

La determinación de las relaciones existentes entre las variables se manifestó desde el principio como una cuestión fundamental. En muchas ocasiones el problema está bien estructurado y el experto sabe determinar directamente un modelo gráfico. Sin embargo, es más habitual que no se conozcan, al menos en forma total, las relaciones de influencia entre los elementos que intervienen. Las Redes Bayesianas tienen numerosas aplicaciones en distintos ámbitos: bioinformática (redes de regulación genética, estructura de proteínas, expresión genética) (Friedman, 2000), etc.

En este trabajo nos centraremos en aplicaciones de los modelos gráficos probabilistas en educación, enseñanza, y aprendizaje. Uno de los problemas básicos que ha recibido más interés es investigar las razones por las que un estudiante abandona sus estudios. (Lykourantzou, 2009) en su publicación usa un método de predicción de deserción en los cursos de e-learning, basado en tres técnicas populares de aprendizaje automático: redes neuronales feedforward, máquinas de soporte vectorial y métodos de ARTMAP difuso simplificado. Puesto que una sola técnica puede fallar para clasificar con precisión a algunos estudiantes de e-learning, mientras que otro puede tener éxito, tres sistemas de toma de decisiones, que se combinan de diferentes maneras los resultados de las tres técnicas de aprendizaje automático fueron probadas.

(Dekker, 2009) comparan distintos modelos para predecir las tasas

de abandono durante el primer semestre de los estudios de grado en la Universidad de Eindhoven. Utilizan árboles de clasificación, Naive bayes, regresión logística, bosques de árboles, obteniendo unas tasas de acierto entre el 75% y el 80%.

En la misma línea podemos citar el trabajo de (Porcel, 2010) en el que se analiza la relación del rendimiento académico de los alumnos ingresantes a la Facultad de Ciencias Exactas y Naturales y Agrimensura de la Universidad Nacional del Nordeste (FACENA-UNNE) en Corrientes, Argentina, durante el primer año de carrera con las características socioeducativas de los mismos. Se ajustó un modelo de regresión logística binaria, el cual clasificó adecuadamente el 75% de los datos. Entre las variables más relevantes para explicar el rendimiento académico se encuentran el título secundario obtenido, la carrera elegida y el nivel educacional alcanzado por la madre.

Dentro de los problemas de diagnóstico (García, 2009) propone un modelo basado en redes bayesianas para determinar el estilo de aprendizaje de cada estudiante. La red se construye a partir de la información proporcionada por expertos (docentes). Dicho modelo es validado con alumnos obteniendo un alto grado de precisión.

Algunas aplicaciones van enfocadas al estudio de medidas de rendimiento colectivo, en lugar de centrarse en un estudiante. En esta línea podemos citar el trabajo de (Oviedo Bayas, 2016) en el que se propone el uso de modelos

gráficos probabilísticos en el campo de la enseñanza para la realización del diagnóstico de los estudiantes y poder predecir su comportamiento. En los últimos años ha cobrado gran importancia el uso de indicadores para describir el perfil de las Universidades españolas en términos tanto académicos como investigadores y económicos. Estos indicadores son utilizados para tomar decisiones de gran importancia, llegando a afectar incluso a aspectos de financiación. Sin embargo, el número de indicadores a veces es excesivo, lo que aumenta el riesgo de redundancia y disfuncionalidad. Los modelos gráficos permiten obtener, a primera vista, las principales relaciones entre las variables a considerar. La metodología propuesta se aplica a un caso práctico, mostrando que es una herramienta útil para ayudar en la toma de decisiones en la elaboración de políticas basadas en indicadores de rendimiento. Esta tarea requiere el manejo de un alto número de variables de distintas naturalezas (cualitativas y cuantitativas), que pueden tener una compleja estructura de dependencias.

## MATERIALES Y MÉTODOS

El aprendizaje estructural se puede definir como el proceso que a partir de un conjunto de datos  $D$ , compuesto por una serie de valores de un conjunto de variables  $X$  para  $n$  casos diferentes, encuentra una topología de red Bayesiana  $B$  que consigue un mejor ajuste al conjunto de datos  $D$  (Friedman N. a., 1999). El proceso de aprendizaje estructural consiste en la selección de

una red sobre un conjunto de dimensión exponencial de estructuras de redes. Por tanto, este proceso lleva asociada una gran dificultad en términos del número de casos requeridos para el muestreo de las posibles redes y el tiempo requerido para la optimización. El proceso de aprendizaje estructural se fundamenta en la utilización de una estrategia de búsqueda para explorar todas las posibles topologías de redes Bayesianas. Como el número de posibles grafos crece de forma exponencial con el número de nodos, no se puede realizar una búsqueda exhaustiva sobre todos los posibles grafos, sino que se utilizan diferentes algoritmos de aprendizaje estructural. Los algoritmos de aprendizaje estructural se pueden clasificar en dos grupos de aproximaciones (Heckerman, 1995).

Nuestro estudio se llevará a cabo para definir otra forma de aprendizaje estructural basado en la búsqueda de estructuras a través de modelos meta-heurísticas evolutivas. Una de las primeras ideas de en este sentido fue la propuesta de un algoritmo genético con codificación binaria presentado en (Kuijpers, 1996). En este modelo un cromosoma codifica una matriz de adyacencia  $G$  que representa la estructura de una red bayesiana y cada hijo es corregido por un operador que elimina ciclos si llegan a existir.

Existe también el algoritmo de Optimización basada en Colonias de hormigas (De Campos, 2002) propuesto por L.M. de Campo donde cada camino de una hormiga  $k$  es representado como

un grafo, de tal forma que cada arista añadida en el camino, representa la unión de dos nodos cualquiera  $(x_i, x_j)$ , y la probabilidad con la cual una hormiga  $k$  añade una arista en su camino, depende de una función heurística que mide la deseabilidad de dicho estado.

Una meta-heurística recientemente introducida en la comunidad científica es la Optimización basada en Malla Variables (Variable Mesh Optimization, VMO). VMO (Puris, 2012) 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.

Este modelo ha sido probado en dominios reales con muy buenos resultados. De aquí el interés por probar su desempeño en el problema del aprendizaje estructural de redes

bayesianas. Para ello debemos de proponer una estructura del algoritmo para ambientes discretos, como es el caso del problema en cuestión.

Variable Mesh Optimization (VMO) (Oviedo, 2016) es una población metaheurística donde las soluciones se distribuyen como una malla en el espacio  $M$ -dimensional. Esta malla contiene  $P$  soluciones candidatos  $(n_1; n_2; \dots; n_p)$ , en este contexto llamadas nodos, y con una representación según el problema de dos operaciones: los procesos de expansión y contracción son ejecutados por VMO para explorar el espacio de búsqueda. Estos procesos son utilizados para introducir un equilibrio adecuado entre la explotación y exploración en algoritmo VMO. Ambos elementos, expansión de malla y contracción, usan una distancia mínima  $\xi$  recomendada entre los individuos (Oviedo, 2016). Esta distancia, se reduce durante la búsqueda y es responsable de mantener la diversidad en la malla. En las subsecciones siguientes, cada fase del algoritmo es descrito y los parámetros están identificados.

**1. Operación de expansión de malla:** la operación de expansión se ejecuta para mover la población por el espacio de búsqueda utilizando diferentes estrategias para crear nuevas soluciones. Cada nodo creado en el proceso de expansión se introduce en una malla temporal  $M^{\text{temp}}$  que se inicializa al comienzo de cada iteración. El algoritmo comienza con una fase inicial, donde la malla inicial  $M$  se crea con  $P$  (parámetro que identifica el número

de nodos de la malla para comenzar cada iteración) soluciones generados al azar con la distribución uniforme. A continuación, se describe cada paso de esta operación:

- **Fase 1.** (Exploración local): en este paso, el vecindario de cada nodo  $(n_i)$  de  $M$  es explorado. Primero, el  $k$  (parámetro que identifica el tamaño del vecindario) nodos más cercanos (usando la función de distancia o similitud) para  $(n_i)$  son identificados y el mejor nodo (fitness) en el vecindario se selecciona como el extremo local  $(n_i^*)$ . En esta fase, solo cuando  $n_i^*$  es óptimo local (mejor fitness que  $n_i$ ) se genera un nuevo nodo usando  $n_i$  y  $n_i^*$  a través de una función que permite combinar funciones.
  - **Fase 2.** (Exploración global): Este paso tiene como objetivo explorar en dirección de un óptimo global (nodo de la malla actual con mejor fitness,  $n_g$ ).  $P-1$  nuevas soluciones son generadas combinando cada nodo  $n_i$  de la malla con  $n_g$ .
  - **Fase 3.** (Exploración selectiva) En este paso, la frontera interior (más similar a los otros nodos de la malla,  $n_u$ ) y la frontera exterior (la menos similar a los otros nodos de la malla,  $n_s$ ) son identificadas usando una matriz de similitud. Entonces, tanto  $n_u$  como  $n_s$  son transformadas para crear dos nuevos nodos.
- 2. Proceso de contracción de malla:** la operación de contracción es utilizada para seleccionar los

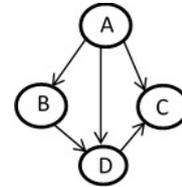
nodos más representativos entre las mallas  $M$  y los nuevos nodos  $M^{temp}$ , eso se mantendrá en la malla  $M$  para la próxima iteración. La idea es mantener los nodos que no están demasiado cerca de los mejores nodos, manteniendo un mínimo distancia entre los nodos seleccionados. En lo siguiente, un operador de compensación adaptativa se describe con el fin de obtener el nodo más representativo:

- **Fase 4.** Todos los nodos ( $M$  u  $M^{temp}$ ) se ordenan dependiendo de su fitness (primero del mejor fitness).
- **Fase 5.** Se calcula la similitud entre cada par de nodos. Si este valor es más que un valor  $\xi$ , entonces el peor nodo es borrado. El valor de  $\xi$  se calcula en función del problema y cambia dependiendo de la iteración. Este paso reduce el número de nodos a  $B$  nodos.
- **Fase 6.** Los nodos con mejor fitness son seleccionados como  $M$  para la próxima iteración. Si el tamaño  $M$  alcanza su valor mínimo la malla se completa con nuevos nodos de manera aleatoria.

En este proceso, los nodos con mejor fitness y más diferentes tienen una mayor probabilidad de ser seleccionados para la próxima iteración.

El algoritmo VMO se puede aplicar al aprendizaje estructural de una red bayesiana (BayesVMO), pero algunos elementos deben adaptarse a las condiciones del problema:

- **Representación**, representamos el nodo de la malla como un conjunto de arcos dirigidos  $(a, b)$ , donde  $a, b$  son variables de la BN y el orden define la dirección del arco (ver Figura 1).



Node1 = {(A,B); (A,C); (A,D); (B,D)};

**Figura 1:** Representación de un BN como conjunto de arcos

**Fuente:** (Oviedo Bayas, 2016)

- **Medida de similitud**, para calcular la similitud entre dos BN, aplicamos la operación de intersección entre los conjuntos de arco  $(N_1 \cap N_2)$  y la cardinalidad del conjunto de intersección representa la similitud:

$$\text{Similitud}_{N_1, N_2} = \text{Card}(N_1 \cap N_2)$$

Esta similitud se usa en la fase 3 del algoritmo para crear una matriz de similitud y encontrar los BNs más (la suma de las similitudes con otras redes es la más alta) y menos (la suma de las similitudes con otras redes es la más baja) nodos similares representativos.

- **Generación de BN**, cada vez que se crea una BN, debemos verificar que no haya ciclos. En caso de que exista un ciclo, uno de sus arcos se selecciona aleatoriamente y su dirección cambia o se elimina (aparece el arco en ambas direcciones). Este proceso se repite hasta que se eliminen todos los

ciclos. Para generar un nuevo BN, se usan dos operaciones:

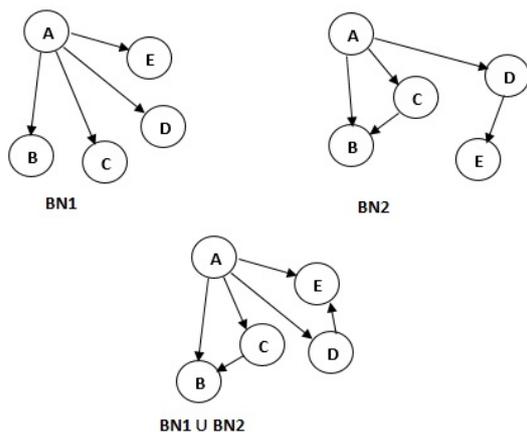
**Generación aleatoria:** cada arco se genera aleatoriamente y se agrega al BN si las dos restricciones siguientes son verdaderas: el arco no incorpora un ciclo en el BN y no está dirigido a la variable clase. El proceso se repite hasta que se genera un ciclo, en cuyo caso el arco no se agrega y el proceso se detiene.

**Generación a partir de dos BNs:** para generar un nuevo BN a partir de dos BNs, se usan dos operaciones en nuestros experimentos. La operación unión y la operación unión de diferencias.

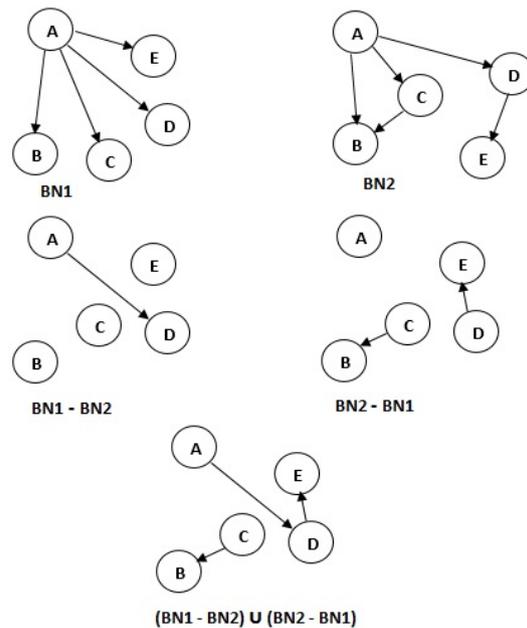
La operación Unión obtiene una nueva BN con todos los arcos de las dos BN operadas. Por otro lado, la unión de las diferencias genera una nueva BN con los arcos presentes en una de las BN pero no en la otra. Las dos operaciones se presentan en la figura

$$BN_{new} = (BN_1 \cup BN_2)$$

$$BN_{new} = (BN_1 \cap BN_2) \cup (BN_2 \cap BN_1)$$



**Figura 2:** Operación Unión entre BN1 y BN2  
Fuente: (Oviedo Bayas, 2016)



**Figura 3:** Operación Unión de diferencias entre BN1 y BN2

Fuente: (Oviedo Bayas, 2016)

La cota de distancia o valor de separabilidad entre soluciones, es otro elemento que se redefine. Donde  $n$  representa el número de variables o características del problema, la iteración actual del algoritmo VMO e  $l$  el número máximo de iteraciones, se incorporan la variabilidad de  $\xi$ . En este caso es el siguiente:

$$\xi = \begin{cases} 0.8 * n & \text{si } r < 1/4 * l \\ 0.7 * n & \text{si } r < 2/4 * l \\ 0.6 * n & \text{si } r < 3/4 * l \\ 0.5 * n & \text{otros casos} \end{cases}$$

La Figura 4 muestra el esquema BayesVMO. Para obtener el valor de fitness, se utiliza la precisión de clasificación con una validación cruzada de 10 veces. No utilizamos las métricas locales como, K2, BIC, MLD u otros tipos de métricas de puntaje.

```

Init Generate initial Mesh  $M$  with  $P$  random BNs.
 $M^{temp} \leftarrow M$ .
Identify current best BN  $n_g \in M$ .

Phase 1 For each  $n_i$  in the mesh that it is not a local optimum:


- Obtain its neighbour with best fitness  $n_i^*$ .
- $n_i^{ph1} = combination_1(n_i, n_i^*)$  using equation (5).
- $M^{temp} \leftarrow M^{temp} \cup n_i^{ph1}$ .

Phase 2 For each  $n_i$  in the mesh:


- $n_i^{ph2} = combination_2(n_i, n_g)$  using equation (6).
- $M^{temp} \leftarrow M^{temp} \cup n_i^{ph2}$ .

Phase 3 Expand the mesh:


- Select  $n_s, n_u \in M$  as the most and least similar BNs in  $M^{temp}$ .
- $n_s^{ph3} = change(n_s)$  and  $n_u^{ph3} = change(n_u)$ , randomly select an arc and change its direction.
- $M^{temp} \leftarrow M^{temp} \cup n_s^{ph3} \cup n_u^{ph3}$ .

Clearing Remove worst solutions from  $M^{temp}$ .


- Sort  $M^{temp}$  by their fitness.
- For each  $n_i, n_k \in M^{temp}$  calculate its similarity using equation (4).
- Remove  $n_k$  if  $similarity(n_i, n_k) \geq \xi_j$  and  $(fitness(n_i) \geq fitness(n_k))$ , where  $\xi_j$  is defined by equation (7).

New  $M$  Create next  $M$  from  $M^{temp}$ .


- If  $|M| < |M^{temp}|$   $M \leftarrow$  bests solutions from  $M^{temp}$ .
- If  $|M| > |M^{temp}|$   $M \leftarrow M^{temp} \cup$  random solutions.
- $M^{temp} \leftarrow M$ .
- Identify current best solution  $n_g \in M$ .

End Iteration Return Phase 1 if not stopping criterion.

```

**Algoritmo 1.** Funcionamiento VMO para aprendizaje de BN  
**Fuente:** (Oviedo Bayas, 2016)

## RESULTADOS

El punto de referencia utilizado para este estudio experimental está compuesto

por 14 conjuntos de datos del repositorio UCI ML [Uci (2013)]. La descripción de estos conjuntos de datos se presenta en la Tabla 1.

**Tabla 1.** Conjunto de datos

Conjunto de datos	Atributos discretos	Atributos continuos	Clases	Instancias
Mammographic	4	1	2	961
Lung-cancer	56	0	3	32
Hepatitis	13	6	2	155
E colic	0	7	8	336

Conjunto de datos	Atributos discretos	Atributos continuos	Clases	Instancias
Breast-cancer-w	10	0	2	683
Contact-lenses	4	0	3	24
Hayes-roth-m	4	0	3	132
Monk	1	6	0	2
Vote	16	0	2	300
Balance-scale	4	0	3	625
Tic-tac-toe	9	0	2	958
Iris	0	4	3	150
Labor	8	8	2	57
Soybean	35	0	19	683

Fuente: (UCI Machine Learning Repository, s.f.)

Los valores para los parámetros de VMO son: tamaño de la red inicial  $P = 12$ , tamaño de vecindad  $k = 3$ , número de iteraciones del algoritmo  $l = 10000$ .

K2, TAN, NBayes, SNB y otros descritos por María del Carmen Chávez M. en su tesis doctoral (Chávez, 2008).

## DISCUSIÓN

Se realiza un estudio comparativo con otros algoritmos. Los clasificadores bayesianos con los que se compara son:

Los experimentos se han llevado a cabo siguiendo la especificación del punto de referencia. La Tabla 2 muestra los valores medios de 25 ejecuciones independientes de cada algoritmo.

Tabla 2. Comparación con otros clasificadores

Datos	ByNet	BChaid	BPSO	BN K2	BN TAN	NBayes	SNB	BVMO
Mammographic	81.89	83.14	82.62	82.41	81.27	83.25	79.29	84.39
Lung-cancer	78.13	75.00	78.13	71.88	65.63	75.00	46.87	78.13
Hepatitis	85.16	86.45	83.87	84.52	85.16	85.16	87.74	93.55
E colic	67.26	85.12	84.52	85.12	84.82	85.42	62.79	76.49
Breast-cancer-w	90.78	97.51	97.36	97.36	95.31	97.36	87.83	97.85
Contact-lenses	87.50	83.33	83.33	70.83	66.67	70.83	66.66	95.83

Datos	ByNet	BChaid	BPSO	BN K2	BN TAN	NBayes	SNB	BVMO
Hayes-roth-m	57.58	81.06	74.24	72.73	67.42	80.30	57.57	84.09
Monk	72.58	70.97	99.19	79.03	95.97	77.42	57.25	97.58
Vote	94.33	91.67	92.67	91.33	93.67	89.67	94.03	93.79
Balance-scale	63.52	92.16	93.92	92.16	92.96	92.16	82.56	86.72
Tic-tac-toe	70.25	72.96	72.65	76.62	76.83	69.62	73.06	75.68
Iris	95.33	94.00	95.33	94.00	94.67	94.00	92.00	98.00
Labor	84.21	87.72	89.47	91.22	89.47	89.47	90.00	95.00
Soybean	68.08	93.11	89.16	94.58	94.58	92.97	99.87	100.00

Fuente: (Oviedo Bayas, 2016)

La prueba de Iman-Davenport es similar a la prueba de Friedman y se usa para identificar las diferencias significativas en un grupo de algoritmos (más de dos). Si las diferencias existentes son significativas, luego se aplica la prueba de Holm para comparar el algoritmo de control (el mejor ranking) con los restantes.

La Tabla 3 muestra los resultados de la prueba de Iman-Davenport donde la hipótesis de igualdad ha sido rechazada (valor de prueba estadística, 3.929, es mayor que el valor crítico, 2.032) detectando diferencias significativas en el grupo.

Tabla 3. Resultados de la prueba de Iman-Davenport

Valor de la prueba	Valor Crítico	Hipótesis
3.929	2.032	Rechazada

Fuente: (Oviedo Bayas, 2016)

A continuación, se realiza la prueba de Holm utilizando BayesVMO como algoritmo de control. La Tabla 4 muestra los resultados de esta prueba donde se puede visualizar que los resultados obtenidos por BVMO son

significativamente superiores en relación a los otros clasificadores. Se concluye esto en virtud de que los p-valores son menores a los valores correspondientes de  $\alpha/i$ .

Tabla 4. Resultados de la prueba de Holm teniendo BVMO como algoritmo de control

Algoritmo	Z	p-valores	$\alpha/i$	Hipótesis
SNB	4.243	2.19E-5	0.0062	Rechazado
NB	3.864	1.11E-4	0.0071	Rechazado

Algoritmo	Z	p-valores	$\alpha/i$	Hipótesis
ByNet	3.795	1.47E-4	0.0083	Rechazado
BN TAN	2.622	0.0087	0.01	Rechazado
BN K2	2.415	0.0157	0.0166	Rechazado
BayesChaid	2.242	0.0243	0.025	Rechazado
BayesPSO	2.001	0.0453	0.05	Rechazado

Fuente: (Oviedo Bayas, 2016)

## CONCLUSIONES

En este trabajo, hemos adaptado la optimización de malla variable (VMO) metaheurístico al aprendizaje estructural de los clasificadores de redes bayesianas (BVMO). Hemos utilizado como puntaje métrico la precisión de clasificación y las operaciones de la VMO se han adaptado a este problema. Entonces, hemos llevado a cabo experimentos utilizando 14 conjuntos de datos del repositorio UCI como punto de referencia. Los resultados de BVMO se han comparado con 8 clasificadores del estado del arte y los análisis estadísticos mostraron que el BVMO alcanza lo mejor resultados.

Para el futuro, planeamos aplicar la metaheurística VMO al aprendizaje genérico de las redes bayesianas, cuando el objetivo es estimar una probabilidad conjunta distribución para las variables problema, en lugar de construir un clasificador.

## AGRADECIMIENTOS

Esta investigación fue apoyada por el proyecto FOCICYT de la Universidad Técnica Estatal de Quevedo

## REFERENCIAS

- Chávez, M. (2008). Modelos de redes bayesianas en el estudio de secuencias genómicas y otros problemas biomédicos. Villa Clara: Universidad Martha Abreu.
- Cooper, G. a. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, 309-347.
- De Campos, L. a.-L. (2002). Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning*, 291-311.
- Dekker, G. a. (2009). Predicting students drop out: A case study. En *Educational Data Mining 2009*.
- Friedman, N. a. (1999). Learning Bayesian Network Structure from Massive Datasets: The Sparse Candidate Algorithm. En *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (págs. 206-215). San Francisco: Morgan Kaufmann Publishers Inc.

- Friedman, N. a. (2000). Using Bayesian networks to analyze expression data. *Computational Biology*, 601-620.
- García, F. (2009). *Modelos bayesianos para la clasificación supervisada: aplicaciones al análisis de datos de expresión genética*. Granada: Universidad de Granada.
- Heckerman, D. a. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 197-243.
- Kuijpers, P. L. (1996). Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 912-925.
- Lykourantzou, I. a. (2009). Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education*, 950-965.
- Nielsen, J. D. (2009). Supervised classification using probabilistic decision graphs. *Computational Statistics & Data Analysis*, 1299-1311.
- Oviedo Bayas, B. W. (2016). *Modelos gráficos probabilísticos aplicados a la predicción del rendimiento en educación*. Granada: Universidad de Granada.
- Oviedo, B. a. (2016). Learning Bayesian network by a Mesh of points. *IEEE World Congress on Computational Intelligence (IEEE WCCI 2016)*.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.
- Porcel, E. a. (2010). Predicción del rendimiento académico de alumnos de primer año de la FACENA (UNNE) en función de su caracterización socioeducativa. *Revista Electrónica de Investigación Educativa*, 1-21.
- Puris, A. a. (2012). Variable mesh optimization for continuous optimization problems. *Soft Computing*, 511-525.
- UCI Machine Learning Repository. (s.f.). <https://archive.ics.uci.edu/ml/about.html>.