Universidad Central “Marta Abreu” de Las Villas

Facultad de Matemática, Física y Computación

Trabajo para optar por Título de

Licenciatura en Ciencia de la Computación

Algoritmo para optimizar la topología en un Mapa Cognitivo Difuso

Autor

Ricardo Pérez García

Tutores

Lic. Gonzalo Nápoles Ruiz

Lic. Isel Grau García

2014

DICTAMEN

Hago constar que el presente Trabajo para optar por Título de Licenciado en Ciencia de la Computación ha sido realizado en la facultad de Matemática, Física y Computación de la Universidad Central “Marta Abreu” de Las Villas (UCLV) como parte de la culminación de los estudios de Licenciatura en Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución para los fines que estime conveniente, tanto de forma total como parcial y que además no podrá ser presentado en eventos ni publicado sin la previa autorización de la UCLV.

_____

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y que el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____ _____

Firma del Tutor Firma del Tutor

Lic. Gonzalo Nápoles Ruiz Lic. Isel Grau García

RESUMEN

Los Mapas Cognitivos Difusos (MCD) son una técnica de computación blanda que constituye una prometedora metodología de modelado, especialmente útil para modelar y simular sistemas dinámicos. De forma general éstos se pueden definir como grafos pesados dirigidos con retroalimentación, consistentes en nodos y de relaciones causales entre los nodos. Éstos fueron introducidos por Kosko y desde entonces han surgido numerosos algoritmos de aprendizaje supervisados y no supervisados, los cuales han estado principalmente enfocados en la estimación de la causalidad del sistema (matriz de pesos causales) usando datos históricos y/o la intervención de expertos, mostrando buenos resultados. Estos algoritmos de aprendizaje pueden ser catalogados en tres grupos: algoritmos hebbianos, algoritmos poblacionales y algoritmos híbridos. A pesar de su éxito, estos poseen ciertas limitantes. Es válido mencionar que hasta donde el autor tiene conocimiento no existen algoritmos de aprendizaje para MCD que permitan encontrar el subconjunto mínimo de nodos centrales que describa el sistema modelado de una forma eficiente. Con este objetivo proponemos un algoritmo de aprendizaje discreto basado en técnicas de Inteligencia Colectiva. El comportamiento de esta propuesta es estudiado a través de un caso de estudio real concerniente a la predicción de la resistencia a fármacos de proteínas del VIH.

ABSTRACT

Fuzzy Cognitive Maps (FCM) is a soft computing technique which constitute a promising modeling methodology, especially useful for modeling and simulating dynamic systems. In general, they can be defined as directed weight graphs with feedback, consisting in nodes and causal relations among the nodes. Since its introduction, many supervised and not supervised learning algorithms have been proposed, mainly focused in estimated the system causality (causal weight matrix) using historic data and/or expert's intervention, showing good results. These learning algorithms can be classified in three groups: hebbian-based algorithms, population-based algorithms and hybrid approaches. Despite their success, they have some limitations. It is valid to mention that as far as the author knowledge, there not exist learning algorithms for FCM that allows finding the subset of central nodes describing the modeled system in an efficient way. With this goal in mind, we propose a discrete learning algorithm based in Swarm Intelligence techniques. The proposal's behavior is studied through a real study case related to the field of drug resistance prediction in HIV.

TABLA DE CONTENIDOS

INTRODUCCIÓN

La Teoría de los Mapas Cognitivos (MC) fue originalmente propuesta en (Axelrod et al., 1976) introduciendo los MC para representar un conocimiento científico social y describir los métodos que son usados para la toma de decisiones en los sistemas políticos y sociales. Estos representaban limitantes en cuanto a la representación de la semántica de sistemas reales. Para solucionar estas limitantes Bart Kosko introdujo los MCD en (Kosko, 1986, Kosko, 1992) considerando valores difusos en los grados de activación de los conceptos de los mapas cognitivos y grados difusos en las interrelaciones de los conceptos. La metodología de los MCD es una representación simbólica de la descripción y modelado de sistemas complejos, estos describen diferentes aspectos en el comportamiento del sistema en términos de conceptos y conexiones causales. Los conceptos representan variables, estados, objetos, o entidades del problema modelado y las interacciones de los conceptos a través de las relaciones causales representan la dinámica del sistema. De forma general los MCD describen el sistema a través de un grafo dirigido pesado con retroalimentación, mostrando la causa y efecto entre los conceptos y éstos son una forma simple de describir el modelo del sistema y su comportamiento de una manera simbólica explotando el conocimiento acumulado del sistema.

Los MCD han resultado de un gran interés en los investigadores debido a sus características dinámicas y su capacidad de aprendizaje. En la literatura se pueden encontrar diversas propuestas de técnicas de aprendizaje las cuales están dirigidas, según (Papageorgiou, 2012) , en tres direcciones :

- Producción de matrices de pesos utilizando datos históricos.
- Adaptación de las relaciones causa-efecto basándose en la intervención de expertos.
- Producción de matrices de pesos para MCD combinando los datos históricos disponibles y la intervención de los expertos en el dominio de aplicación.

Estas técnicas de aprendizaje son catalogadas según (Papageorgiou, 2012) en tres principales grupos:

- Los algoritmos hebbianos.
- Los enfoques poblacionales.
- Los métodos híbridos.

Se puede apreciar que estos algoritmos de aprendizaje están enfocados principalmente sobre la construcción o actualización de la matriz de pesos causales. Además presentan limitaciones, por ejemplo: los algoritmos hebbianos precisan la intervención del experto y su meta es producir la matriz de

pesos causales basados solamente en el conocimiento del experto (Papageorgiou, 2012) lo cual reduce la usabilidad del sistema final. Los enfoques poblacionales generalmente no requieren la intervención del experto debido a que estos se basan en información histórica del sistema. Evidentemente su desempeño empeora cuando aumenta la cantidad de variables del sistema; y las funciones de evaluaciones heurísticas optimizadas durante el proceso son usualmente multimodales (Nápoles et al., 2013) y tienden a converger prematuramente.

Los sistemas reales a modelar son usualmente muy complejos e incluyen un gran número de variables. Posterior a la estimación de la causalidad del sistema y simulada su dinámica o estudiadas las interacciones entre los conceptos que describen el sistema se hace muy difícil la interpretación del modelo final, lo cual empeora cuando en la construcción del sistema participan múltiples expertos en la selección de las variables. Esto da paso a la incertidumbre en la modelación produciendo así conceptos parcialmente contradictorios en el mapa. Por este motivo, resulta evidente la necesidad de nuevos algoritmos de aprendizaje para computar el subconjunto mínimo de conceptos que mejor describan el sistema investigado.

Es válido mencionar que basado en el trabajo de revisión (Papageorgiou, 2012), es razonable suponer que no existen algoritmos de aprendizaje orientados a solucionar este difícil problema combinatorio desde la perspectiva de los MCD deviniendo la línea de investigación que intentará abordar esta propuesta.

Objetivo general

Desarrollar un algoritmo de aprendizaje supervisado para optimizar la topología de un MCD, utilizando una meta-heurística poblacional implementando el modelo en un ambiente de desarrollo y simulación computacional.

Objetivos específicos

1. Desarrollar un algoritmo de aprendizaje supervisado para ajustar la topología de un MCD como un problema de optimización discreto.
2. Implementar el algoritmo de aprendizaje utilizando una meta-heurística poblacional.
3. Evaluar el desempeño del algoritmo propuesto a través de un caso de estudio real concerniente a la predicción de la resistencia a fármacos de proteínas del VIH.

Tareas de Investigación

1. Realizar un estudio de los principales algoritmos de aprendizaje reportados en la literatura (supervisados y no supervisados) para Mapas Cognitivos Difusos.
2. Modelar el algoritmo de aprendizaje supervisado como un problema de optimización discreto (representación de las soluciones, función objetivo y restricciones).
3. Seleccionar una meta-heurística poblacional que permita solucionar problemas discretos.
4. Implementar el algoritmo resultante en la herramienta de simulación y experimentación *FCM Tool*.
5. Diseñar un experimento estadístico para comprobar la optimización de la topología utilizando un problema de Bioinformática como caso de estudio.

Justificación

Los MCD han sido ampliamente utilizados en varios dominios de aplicación, por ejemplo: problemas de toma de decisiones, análisis de riesgo, embebidos en sistemas de control, categorización de textos, etcétera. De hecho, en los últimos años las investigaciones relacionadas con los MCD se han incrementado considerablemente (Papageorgiou, 2011). La optimización de la topología de MCD a través de la obtención del subconjunto mínimo de variables que describan al sistema completo permite una reducción de la dimensión del MCD y posibilita una mejor interpretación del modelo final. Hasta donde el autor tiene conocimiento la optimización de la topología de un MCD a través de la obtención del subconjunto mínimo de variables que describen al sistema completo de manera eficiente no ha sido tratada en la literatura existente.

Hipótesis de investigación

El algoritmo de aprendizaje supervisado propuesto permite optimizar la topología de un MCD a través de la reducción de los conceptos no relevantes.

Diseño de la Tesis

Después de la Introducción, este trabajo consta de tres capítulos. El primero de ellos propone un acercamiento a la teoría básica de MCD y a los algoritmos de aprendizajes más representativos reportados en la literatura. El segundo capítulo introduce el algoritmo de aprendizaje discreto que permita estimar el conjunto mínimo de nodos centrales en un MCD para la reducción del número de conceptos que caracterizan al sistema modelado. En el tercer capítulo el comportamiento del algoritmo es evaluado a través de un problema real de clasificación. Además de formular conclusiones y recomendaciones, se listan las referencias bibliográficas utilizadas.

CAPÍTULO 1. TEORÍA DE MAPAS COGNITIVOS DIFUSOS

En este capítulo se aborda la teoría de MCD desde su introducción en 1986 por Bart Kosko. Además incluye un análisis sobre el proceso de inferencia. En una segunda parte se introduce una revisión sobre los principales algoritmos de aprendizaje reportados en la literatura.

1.1 Mapas Cognitivos Difusos

La teoría de MC fue originalmente propuesta en (Axelrod et al., 1976), donde el autor se enfocó en los estudios del campo de la política, específicamente para representar un conocimiento científico social. En términos generales, un simple MC puede ser caracterizado por conceptos e influencias causales. De esta manera, los conceptos representan variables, entidades o estados que usualmente describen el comportamiento del modelo para un determinado dominio; mientras que las influencias causales son relaciones entre las variables, quienes tienen un signo (- o +) para expresar una causalidad específica. Por ejemplo:

- Si el signo de una relación es positivo, entonces un incremento o reducción de la variable causa que las variables afectadas por ésta cambien en la misma dirección (ejemplo: un incremento en la variable causa un incremento en las variables afectadas por la misma)
- Si el signo de una relación es negativo, entonces un incremento o reducción de la variable causa que las variables afectadas por ésta cambien en dirección contraria, es inversamente proporcional (ejemplo: un incremento en la variable causa una reducción en las variables afectadas por la misma).

Los MC han sido ampliamente utilizados en ámbitos como la toma de decisiones, simulación de sistemas, economía, administración, robótica, agentes, sistemas dinámicos, ingeniería, análisis de información y adaptación (León et al., 2010) gracias a las bondades del modelo computacional que posibilita una simple representación de los componentes del sistema y sus relaciones; permitiendo diversos niveles de abstracción del problema a través de los conceptos.

A pesar de las ventajas que brindan los MC para manejar el conocimiento del sistema, las capacidades de representación de las causalidades son sumamente limitadas debido a que es complicado definir la semántica de todo el sistema solamente expresando un signo en las relaciones, sin especificar un grado de causalidad.

Para solucionar dichas limitaciones, Bart Kosko expandió la teoría de MC introduciendo los MCD en (Kosko, 1986) al considerar la aplicación de la Lógica Difusa (McNeill and Thro, 1994) en la cuantificación de los nodos que representan conceptos y los arcos que enlazan los nodos mediante

grados de pertenencia y los umbrales difusos. Sin perder generalidad, los MCD pueden ser definidos por un vector de estado A_{1xn} el cual incluye los valores de n conceptos y una matriz de pesos causales W_{nxn} la cual contiene los pesos w_{ij} de las interconexiones causales entre los conceptos.

Las relaciones causales en lugar de usar un signo en cada conexión, se le asignan un peso que denota el grado de causalidad, permitiendo así que una relación pueda ser ahora representada por un término difuso, por ejemplo: bajo, medio, alto. La interpretación de las influencias causales entre los conceptos de un MCD es la misma que un mapa cognitivo. Para dos conceptos C_i y C_j se cumple:

- Si el peso w_{ij} asociado a la conexión entre los conceptos C_i y C_j es mayor que cero, entonces la relación es directamente proporcional; ejemplo: (un incremento (disminución) en el concepto C_i provocará un incremento (disminución) en el concepto C_j de forma proporcional al valor w_{ij}).

- Si el peso w_{ij} asociado a la conexión entre los conceptos C_i y C_j es menor que cero, entonces la relación es inversamente proporcional; ejemplo: (un incremento (disminución) en el concepto C_i provocará una disminución (incremento) en el concepto C_j de forma proporcional al valor w_{ij}).

- Si el peso w_{ij} asociado a la conexión entre los conceptos C_i y C_j es igual (o muy próximo) a cero, entonces no existe relación causal entre los dos conceptos.

Las principales ventajas de la representación del conocimiento a través de relaciones causales fueron listadas por (Huff, 1990):

- Las relaciones causales son una de las principales formas de saber cómo el mundo está organizado.
- La causalidad es la forma primaria *post hoc* de explicación de los eventos.
- Escoger entre opciones alternativas (proceso de decisión) involucra la evaluación causal.

Usualmente las relaciones causales en los MCD siempre involucran cambio y su resultado es siempre la variación en uno o más conceptos.

Desde el punto de vista estructural los MCD son representados como grafos dirigidos pesados con retroalimentación, consistentes de nodos y de arcos pesados (ver Figura 1.1). Además el valor de cada nodo del grafo puede ser expresado como un valor difuso dentro del rango [-1,1] donde los valores cercanos a 1, indican mayor pertenencia, mientras que los valores cercanos a 0 representan incertidumbre, y grados cercanos a -1 tienden a nula pertenencia (Kosko, 1988). Sin embargo en algunos casos es habitualmente usado el rango [0,1] relacionado con el grado presente de un elemento específico

del sistema en un tiempo t, donde los valores cercanos a 1 señalan un grado activo del concepto, mientras que valores próximos a 0, señalan nula pertenencia (Kosko, 1992). En otras palabras el nivel de activación de un concepto está relacionado con el grado de pertenencia al sistema; o puede ser mapeado como un valor binario representando así la activación o no del concepto. Por tanto, cuando la activación de un concepto es un valor no binario entonces una importante regla caracteriza el comportamiento del mapa: mientras mayor valor de activación tenga un concepto, mayor es la influencia del concepto en el sistema.

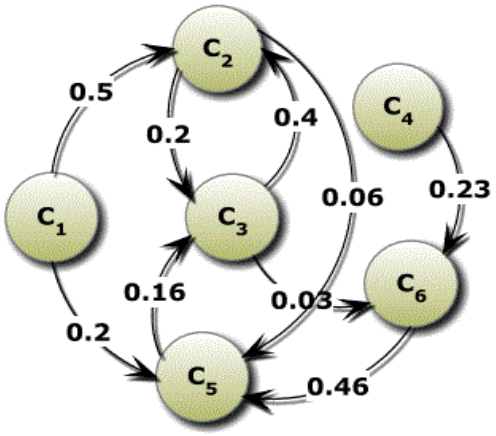


Fig. 1.1 Mapa Cognitivo Difuso.

De forma general los MCD son una combinación de la Redes Neuronales y la lógica difusa que nos permite predecir los cambios en los conceptos representados en los MC (Carvalho and Tomé, 1999). Entre sus principales bondades se encuentra la simple representación de las variables del sistema e interrelación entre ellas a través de las relaciones causales.

1.2 Proceso de Inferencia

Los MCD pueden ser matemáticamente definidos por un vector de estado que denota el nivel de activación de los conceptos y por una matriz de pesos la cual define la interacción de los conceptos según el criterio del experto. El proceso de inferencia o simulación de un MCD consiste en computar, a través del tiempo, el vector de estado correspondiente para una condición inicial.

Sin perder generalidad, los MCD pueden ser definidos por un vector de estado $A_{1\times n}$el cual incluye los valores de n conceptos y una matriz de pesos causales $W_{n\times n}$ la cual contiene los pesos w_{ij} de las interconexiones causales entre los conceptos. Matemáticamente hablando, los estados del sistema son iterativamente actualizados multiplicando la matriz de pesos del MCD por el vector de estado (Kosko, 1992). Por tanto el valor de un concepto C_i está influenciado por los valores de los conceptos C_j

conectados a él y los valores de sus relaciones causales w_{ji}. Esto se resume en la siguiente ecuación (1.1), mostrando el efecto del cambio del valor de un concepto en el mapa completo. Este proceso de inferencia garantiza la retroalimentación del sistema y normalmente es repetido hasta que un punto fijo de atracción (también llamado *patrón oculto*, véase comportamiento en la ecuación 1.2) o el número máximo de tiempos (iteraciones) es alcanzado (Kosko, 1988). Sin embargo, en algunos casos es posible que el mecanismo de inferencia entre en un estado caótico, lo que significa que el modelo MCD continua produciendo diferentes vectores de estados para ciclos sucesivos (Wang et al., 1990).

$$A_i^{(t+1)} = f\left(\sum_{j=1}^{n} w_{ji}\, A_j^{(t)}\right), i \neq j \tag{1.1}$$

Donde $A_i^{(t+1)}$ es el valor (nivel de activación) del concepto C_i en el tiempo $t+1$, w_{ji} es el peso (influencia causal) de la relación entre el concepto C_j y el concepto C_i; y $A_j^{(t)}$ es el valor del concepto C_j en el tiempo t.

$$[A_{i+1} = f(A_i \times W) - f(A_{i+1} \times W)] = 0 \tag{1.2}$$

El proceso de inferencia de un MCD visto desde un punto de vista matricial se realiza por medio de una multiplicación vector-matriz un finito número de veces, donde los vectores de estados A^i iteran a través de la matriz causal W, vea la siguiente ecuación:

$$A_i = f(A_{i-1} * W) \tag{1.3}$$

Donde el vector de estado A_{i-1} del tiempo t_{i-1}, se multiplica por la matriz de pesos causales W y se normaliza con la función umbral $f(x)$ dando como resultado el vector A_i como efecto de esta iteración, el cual se convertirá en causa del $A_{i-ésimo}$ vector representando así la evolución en el tiempo del sistema y garantizando la retroalimentación.

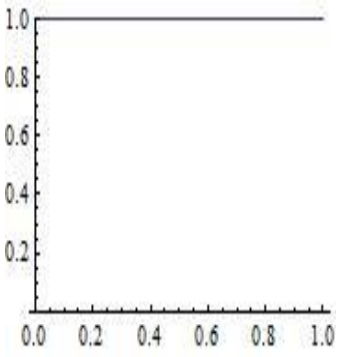
Un importante componente de la ecuación anterior es la función de transformación *f*, la cual es usada para normalizar el valor de activación de los conceptos en un determinado rango. Las funciones de transformación usadas más frecuentemente (vea fig.1) según (Tsadiras, 2008) son:

- La función de signo:
    - El uso de esta función permite que los niveles de activación de cada concepto sean 0 o 1.

- Son apropiadas para problemas altamente cualitativos, donde solamente se requiere representar el incremento o estabilidad de los conceptos que describen los estados del sistema.

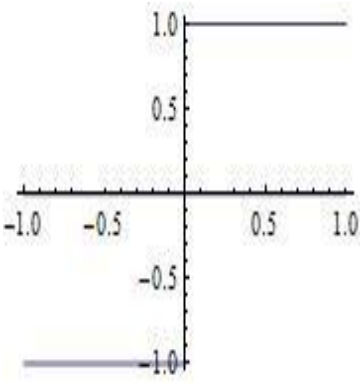
- La función trivalente:
  - Cuando el grado de activación de un concepto sea igual a 1 significa que esta crece, si es igual a -1 decrece y si es igual a 0 se mantiene estable.
  - Son apropiadas para problemas cualitativos, donde el experto puede representar el incremento, decremento o estabilidad de los conceptos que describen los estados del sistema.
- La función sigmoidal:
  - El grado de activación de un concepto varía en el rango [-1,1].
  - Son apropiadas para problemas cualitativos y cuantitativos donde es posible representar el decremento, incremento o estabilidad de los conceptos que describen los estados del sistema.

La selección de esta función está condicionada fuertemente por los requerimientos del sistema modelado, o sea, por el rol que desempeñe la activación de los conceptos en un momento determinado por lo que la selección de una u otra variante puede influir significativamente en el proceso de inferencia.



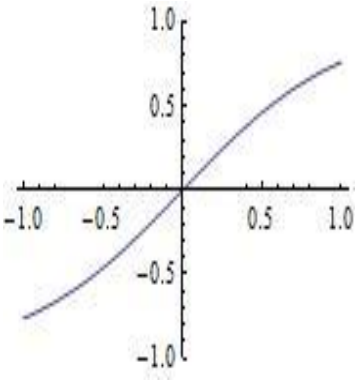
$$f(x) = \begin{cases} 0 & ,x < 0 \\ 1 & ,x \geq 0 \end{cases}$$

Fig. 1.2a. Signo.



$$f(x) = \begin{cases} 1 & ,x > 0 \\ 0 & ,x = 0 \\ -1 & ,x < 0 \end{cases}$$

Fig. 1.2b. Trivalente



$$f(x) = \{\tanh(x)\}$$

Fig. 1.2c. Sigmoidal

Dependiendo de la función de transformación usada en el mecanismo de inferencia varios tipos de simulaciones son posibles. Por ejemplo, aplicando funciones de salida discreta, el modelo eventualmente será capaz de descubrir la existencia de un punto fijo de atracción o puede mantener el ciclo a través de un número finito de vectores de estado. Mientras que usando funciones de salida continuas, el modelo

puede caer en un estado caótico. El uso de una función de transformación específica está determinado por las características del problema. En un estudio más profundo los efectos de la selección de esta función sobre la capacidad de inferencia de los MCD son explorados en (Tsadiras, 2008).

Es válido destacar que una de las principales características del proceso de inferencia en los MCD es la capacidad memorística de estas estructuras. En principio el nivel de activación de un concepto A_i en un tiempo t puede calcularse utilizando la ecuación 1.1, no obstante existe una variante de esta ecuación (vea ecuación 1.4) en la cual se utiliza el estado del propio concepto A_i en el instante de tiempo t_{i-1} como una causa del concepto en el estado actual.

$$A_i^{(t+1)} = f\left(\delta_1 \sum_{j=1}^{n} w_{ji} A_j^{(t)} + \delta_2 A_i^{(t)}\right), i \neq j \tag{1.4}$$

Se puede apreciar que la influencia que un concepto ejerce sobre el otro se observa a través del paso del tiempo (iteraciones) al calcular principalmente los efectos directos, ya que los indirectos poseen un retardo.

De manera general, es pertinente destacar la capacidad de los MCD para simular sistemas dinámicos donde se describe una situación que evoluciona a lo largo del tiempo y procura alcanzar estados de progreso o estabilidad.

1.3 Proceso de Agregación

Generalmente los MCD representan el conocimiento extraído de un experto en un área determinada. Además es posible combinar varios criterios de expertos en la construcción de un MCD, o sea obtener el consenso de varios expertos. Esto se logra a través de un proceso de agregación de varios mapas. Dicho proceso de agregación permite que el mapa resultante sea menos propenso a los efectos indeseables de creencias erróneas o ignorancia logrando así aumentar la fiabilidad del mapa resultante.

El número de expertos que pueden estar involucrados en la modelación del mapa normalmente no tiene restricciones, mientras más expertos estén involucrados más fiabilidad tendrá el sistema resultante (Kosko, 1992).

La posibilidad de agregación de varios MCD en una sola estructura de conocimiento es una importante ventaja sobre otros modelos basados en el conocimiento como por ejemplo Redes Bayesianas o Redes de Petri a la hora de describir un mismo sistema.

Pocos métodos de agregación de MCD han sido propuestos en la literatura. El primero fue propuesto por Bart Kosko en (Kosko, 1986) y está basado en la transformación matemática de las matrices de pesos causales de los mapas. La agregación de múltiples mapas puede ser expresada por el promedio, mediana y promedio pesado de sus conexiones (relaciones causales). Si los mapas involucrados en el proceso de agregación están caracterizados por los mismos conceptos entonces el mapa resultante tendrá dichos conceptos con la aplicación de algunos de los operadores previamente mencionados sobre las relaciones causales.

Aunque usualmente los expertos tienen diferencias de opinión en cuanto a los conceptos involucrados en la modelación, por tanto cuando esto ocurre tienen dos posibilidades:

1. Los mapas no comparten ningún concepto, por tanto la matriz de pesos causales resultante del mapa agregado es construida poniendo cada matriz individual a través de su diagonal principal y luego rellenando las restantes posiciones con cero. La dimensión del matriz resultante es el total de conceptos diferentes de todos los mapas (Aguilar, 2005).
2. Generalmente los mapas tienen conceptos comunes. En este caso es necesario aumentar la matriz de cada mapa añadiendo una nueva columna y fila rellenada con ceros por cada concepto que no aparezca en el mapa. Luego las matrices de los mapas tienen la misma dimensión y se pueden aplicar los operadores vistos anteriormente, la siguiente imagen ilustra este procedimiento para tres mapas que comparten conceptos y teniendo en cuenta el promedio para combinar la causalidad entre los conceptos.

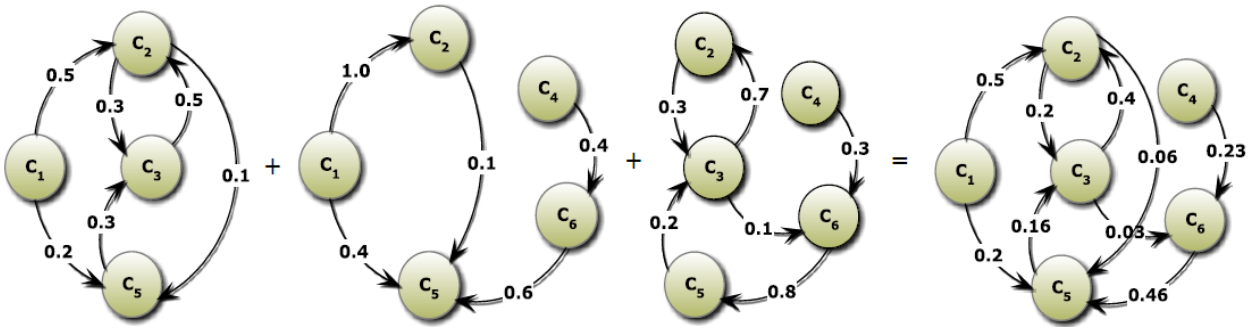


Fig. 1.3 Proceso de Agregación de tres MCD a través del operador promedio.

1.4 Estrategias de desarrollo para Mapas Cognitivos Difusos

Existen dos grupos de técnicas para el desarrollo de MCD. El primer grupo está conformado por los métodos basados en expertos los cuales incluyen técnicas que solamente explotan el conocimiento humano. Durante un largo período de tiempo ésta fue prácticamente la única forma de desarrollar MCD principalmente por la falta de enfoques automáticos o semiautomáticos. El segundo grupo está

conformado por los métodos computacionales. El objetivo de estos es sustituir o ayudar al experto y el aprendizaje de la estructura del modelo de manera automática o semiautomática usando información histórica (Stach et al., 2005a, Stach et al., 2010).

1.4.1 Métodos basados en expertos

El desarrollo basado en expertos de MCD usualmente consiste en los siguientes tres pasos (Kosko, 1986, Khan and Quaddus, 2004):

1. Identificación de los conceptos importantes. La estrategia más intuitiva es crear una lista con todos los conceptos relevantes y remover los más insignificantes.
2. Identificación de las relaciones causales entre esos conceptos. En este paso todas las relaciones directas causa-efecto entre los conceptos seleccionados deben ser identificadas. Usualmente esto es llevado a cabo centrándose en un par de conceptos a la vez y el experto tiene la tarea de definir si existe la relación causa-efecto. En estos dos primeros pasos se define el diseño estructural.
3. Estimación de las fortalezas de las relaciones causales.

El principal desafío en estos métodos es el correspondiente con en el paso 3. Nótese que el número de relaciones causales exhibe un crecimiento cuadrático con respecto al número de conceptos, lo que puede a llegar complicaciones en el desarrollo de los mapas cuando tenga varias docenas de conceptos. En (Kosko, 1986) el valor de cada relación causal(peso) es expresado a través de un número en el intervalo $[-1,1]$. Teóricamente cada peso puede tomar valores infinitos. Por tanto este paso es muy susceptible a la interpretación del experto. Una práctica común para facilitar la estimación de los pesos es primero describir cada relación causal por un término lingüístico y entonces transformar estos términos a valores numéricos, estos pasos son descritos a continuación:

1. Determinar el signo de cada relación.
2. Describir cada relación mediante términos lingüísticos.
3. Mapear los términos lingüísticos a valores numéricos

El uso de expresiones lingüísticas para describir los grados de causalidad en las relaciones permite a los expertos evitar la difícil tarea de especificar el valor numérico preciso de cada relación antes de que un modelo básico sea establecido.

1.4.2 Métodos computacionales

Los métodos computaciones utilizan información histórica disponible para un sistema dado para establecer el modelo del MCD. Los métodos semiautomáticos requieren una intervención puntual del

experto, mientras que los métodos automáticos son capaces de computar el modelo del MCD solamente utilizando la información histórica o sea sin ninguna intervención del experto.

Estos son clasificados en tres grupos fundamentales basados en el paradigma de aprendizaje. En la siguiente sección se profundiza en el estudio de estos métodos.

1.5 Algoritmos de aprendizaje para Mapas Cognitivos Difusos

El objetivo principal de los algoritmos de aprendizaje para MCD ha sido optimizar la matriz de pesos causales, lo que significa el ajuste de las fortalezas de las relaciones entre los conceptos. Estos se encuentran divididos en tres grupos según (Papageorgiou, 2012): algoritmos hebbianos, algoritmos inspirados en meta-heurística y algoritmos híbridos.

1.5.1 Algoritmos de aprendizaje hebbianos

La meta de los algoritmos de aprendizaje hebbianos es producir la matriz de pesos causales basado en el conocimiento experto, lo que conlleva al MCD a converger en un estado de decisión dado o en una región aceptable para el problema en cuestión (Papageorgiou, 2012).

Los algoritmos de aprendizaje hebbianos son no supervisados, por lo cual no requieren que los datos estén etiquetados, o sea, pertenezcan a una clase o rasgo objetivo. Dickerson y Kosko propusieron el primer algoritmo de este tipo para estimar la matriz de pesos causales; Algoritmo Hebbiano Diferencial (DHL) en (Dickerson and Kosko, 1993). Este método está basado en la correlación causal del cambio de los conceptos como se muestra en la siguiente ecuación (Stach et al., 2005a).

$$\dot{W}_{ij} = \dot{A}_i \dot{A}_j - w_{ij} \tag{1.5}$$

Donde $\dot{W}_{ij}$ es el cambio de peso de la relación entre el concepto C_i y el concepto C_j; w_{ij} es el peso actual de la relación entre el concepto C_i y el concepto C_j; y $\dot{A}_i, \dot{A}_j$ son los cambios en el valor de los conceptos C_i y C_j respectivamente.

El proceso de aprendizaje del algoritmo DHL iterativamente actualiza los valores de la matriz de pesos causales hasta que se obtenga una estructura de causalidad deseada o se detenga por un número máximo de iteraciones. El algoritmo asume que cuando ocurra un cambio simultáneo en el valor de dos conceptos, el valor de la relación de los mismos incrementará. Considerando el valor de la variación de un concepto en dos estados sucesivos se tiene:

$$\Delta A_i = A_i(t) - A_i(t-1) \tag{1.6}$$

Es válido mencionar que los valores de dos conceptos C_i y C_j solamente aumentan o disminuyen al mismo tiempo cuando $\Delta A_i\, \Delta A_j > 0$. Si ocurriera $\Delta A_i\, \Delta A_j < 0$ esto significaría que el valor de uno de los conceptos aumenta; mientras el otro disminuye. De forma general cuando $\Delta A_i > 0$ significa que el valor del concepto C_i cambió y por tanto los pesos de las conexiones que se inician desde él cambiarán de acuerdo a la siguiente ecuación:

$$w_{ij}^{(t+1)} = \begin{cases} w_{ij}^{(t)} + \gamma(t)\left[\Delta A_i \Delta A_j - w_{ij}^{(t)}\right] & si\ \Delta A_i \neq 0 \\ w_{ij}^{(t)} & si\ \ \Delta A_i = 0 \end{cases} \tag{1.7}$$

Donde el coeficiente $\gamma(t) = 0.1[1 - (t/1.1q)]$ denota el incremento que modificará los valores causales en el proceso de ajuste. En este caso el parámetro $q \in \mathbb{N}$ debe seleccionarse de forma tal que el peso w_{ij} se mantenga en el rango $[-1,1]$ y de que el coeficiente $\gamma(t)$ sea positivo. Por ejemplo, el valor del parámetro q puede ser el número máximo de iteraciones o generaciones en el aprendizaje.

El principal problema de este método de aprendizaje es que solamente se tiene en cuenta el efecto de la fortaleza de la relación causal entre dos conceptos sin tener en consideración la influencia del resto de los conceptos (Papageorgiou, 2012). Para solucionar esta limitante en (Huerga, 2002) se introdujo una versión mejorada del DHL llamada Algoritmo Diferencial Balanceado (BDA) en la cual éste añadió nuevas reglas para actualizar los pesos de las relaciones causales como se puede observar en la siguiente ecuación:

$$w_{ij}(t+1) = \left\{ \begin{array}{lll} w_{ij}(t) + \dfrac{A_i}{q} & , & si\ i = j \\ w_{ij}(t) + \gamma(t)\left[\dfrac{\dfrac{\Delta A_i}{\Delta A_j}}{s_1}\right] - w_{ij}(t) & , & si\ i \neq j\ , \Delta A_i \Delta A_j > 0 \\ w_{ij}(t) + \gamma(t)\left[\dfrac{-\dfrac{\Delta A_i}{\Delta A_j}}{s_2}\right] - w_{ij}(t) & , & si\ i \neq j\ y\ \Delta A_i \Delta A_j < 0 \end{array} \right\} \tag{1.8}$$

$$s_1 = \sum_{\substack{k=1 \\ \Delta A_i \Delta A_k > 0}}^{n} \frac{\Delta A_i}{\Delta A_k} \qquad s_2 = \sum_{\substack{k=1 \\ \Delta A_i \Delta A_k < 0}}^{n} \frac{\Delta A_i}{\Delta A_k}$$

El proceso de aprendizaje de BDA cuando actualiza los pesos de las relaciones causales entre dos conceptos tiene en cuenta los valores de los conceptos que cambian al mismo tiempo (misma iteración), lo cual significa que el $w_{ij}(t+1)$ toma en consideración no solo los cambios ocurridos en ΔA_i y ΔA_j sino que además considera los cambios en los restantes conceptos que ocurren en la misma iteración. Sin embargo este algoritmo de aprendizaje solo es aplicable a MCD binarios (Papageorgiou, 2012).

Por otra parte, el Aprendizaje Hebbiano No-Lineal (NHL) está basado en una extensión no lineal básica de la ley Hebbiana y fue propuesto en (Papageorgiou et al., 2003). Este posee un enfoque semiautomático debido a que necesita una intervención inicial de los expertos referente a tres aspectos esenciales (Papageorgiou, 2012):

- Sugerencia del conjunto de conceptos (conceptos de salida y conceptos de entrada).
- Especificación del rango de valores que puede tomar un concepto.
- Especificación del signo de cada conexión.

La idea principal de este método consiste en la forma en que se actualizan los pesos sugeridos inicialmente por los expertos teniéndose en cuenta los tres aspectos mencionados anteriormente. Es válido mencionar que la especificación del signo de cada conexión diferente de cero realizada por el experto está de acuerdo con su interpretación física (Stach et al., 2005a). Estos cambios en el proceso de actualización de los pesos permite que la estructura del modelo que fue propuesta por los expertos se conserve durante el proceso de aprendizaje manteniendo su interpretación física (Papageorgiou, 2012). Su principal desventaja es que necesita del criterio inicial de los expertos.

Posteriormente fue propuesto el Algoritmo Hebbiano Activo (AHL) como otro intento por estimar la matriz de pesos causales en (Papageorgiou et al., 2004). Este método está estructurado en un procedimiento de siete pasos basado en la teoría de aprendizaje hebbiana. Dicho procedimiento es usado para actualizar los pesos en cada iteración basado en tres factores principales (Papageorgiou, 2012):

- Definición del conjunto de conceptos a través de expertos.
- Definición de la estructura inicial de las relaciones causales.
- Definición de la secuencia de activación de los conceptos.

Conviene destacar que la principal desventaja de este método es su dependencia de la intervención del experto.

Recientemente ha sido propuesta una versión mejorada del algoritmo de aprendizaje NHL denominada Aprendizaje Hebbiano No-Lineal dirigido por dato (DDNHL) en (Stach et al., 2008). Éste está basado en

el mismo principio de aprendizaje de NHL aunque toma ventaja de los datos históricos y usa los conceptos de salida para mejorar la calidad del aprendizaje (Papageorgiou, 2012).

1.5.2 Algoritmos de aprendizaje basados en meta-heurísticas

La meta de los algoritmos de aprendizaje basados en meta-heurísticas es computar la matriz de pesos causales teniendo como base los datos históricos que mejor se adaptan a la secuencia de vectores de estado o patrones (Papageorgiou, 2012).

En los algoritmos de aprendizaje basados en meta-heurísticas usualmente su principal objetivo es producir matrices de pesos causales óptimas para modelos de MCD que describen problemas específicos (Papageorgiou, 2012). En un primer intento Koulourioti propuso en (Koulouriotis et al., 2001) un esquema para el aprendizaje de MCD basado en una estrategia genética (GS). En este método el proceso de aprendizaje está basado en una colección de pares de entradas y salidas. Las entradas son definidas como los valores del vector inicial de estados, mientras las salidas son definidas como los valores del vector final de estados. La función de minimización del error utilizada por este método se muestra a continuación:

$$f_{error} = \sum_{k=1}^{K} \sum_{i=1}^{N} |{O_i}^e - {O_i}^r| \tag{1.9}$$

Donde K es el número de instancias de la base, ${O_i}^e$ es la salida estimada por el algoritmo para el i-ésimo concepto y ${O_i}^r$ es la respuesta real del i-ésimo concepto.

La principal desventaja de este método consiste en que se requieren múltiples secuencias de vectores de estados en el proceso de aprendizaje, las cuales son difíciles de obtener en problemas de la vida real.

Posteriormente se propuso la aplicación de Optimización basada en Bandadas de Partículas (PSO, por sus siglas en inglés) para el aprendizaje de la estructura de los MCD en (Parsopoulos et al., 2003) apoyado en datos históricos para converger en un estado final deseado (Papageorgiou, 2012) donde la salida del sistema es conocida por conceptos decisión (CD). Un experto establece los límites de activación de cada CD: $A_i^{\min} \leq A_i^{DC} \leq A_i^{\max}$.

PSO es un algoritmo cuyo proceso de búsqueda está basado en la transformación y mantenimiento de una población de individuos. Su principal objetivo es mejorar la calidad del modelo resultante a través de la minimización de la función objetivo (vea ecuación 1.10) durante el proceso de búsqueda para obtener una matriz de pesos que minimice las diferencias entre la respuesta inferida por el mapa y

las respuesta conocida para un vector de activación inicial, siempre y cuando satisfaga las restricciones de activación impuestas por los expertos.

$$f = \sum_{i=1}^{n} H(A_i^{\min(\mathrm{DC})} - A_i^{DC}) \left| A_i^{\min(\mathrm{DC})} - A_i^{DC} \right| + \sum_{i=1}^{n} H(A_i^{DC} - A_i^{\max(\mathrm{DC})}) \left| A_i^{DC} - A_i^{\max(\mathrm{DC})} \right| \quad (1.10)$$

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

La principal desventaja de este método radica en la necesidad de intervención de los expertos.

Luego Khan y Chong introdujeron un esquema de aprendizaje diferente de MCD en (Khan and Chong, 2003) basado en algoritmos genéticos (GA). El cual en vez de enfocarse en el aprendizaje de la estructura del modelo del MCD se enfoca en determinar el vector inicial de estados (condiciones iniciales) que conduce el sistema a un estado de atracción deseado o un ciclo limitado (Stach et al., 2005a). Este tipo de análisis es especialmente útil en sistemas de toma de decisiones.

Recientemente fue propuesto por Stach en (Stach et al., 2005b) un esquema de aprendizaje automatizado para la estructura de MCD basado en una extensión de GA llamada Algoritmos Genéticos con Codificación Real (RCGA). Este método maximiza la función objetivo (vea la siguiente ecuación) reduciendo una función de error (vea ecuación 1.12).

$$f = \frac{1}{1 + \alpha * error} \quad (1.11)$$

Donde α denota el coeficiente de calidad de la función.

$$error = \frac{1}{n(K-1)} \sum_{k=1}^{K} \sum_{i=1}^{n} \left| A_i(t) - A_i^p(t) \right| \quad (1.12)$$

Donde K es el número de instancias, n el número de conceptos, $A_i(t)$ es la respuesta del sistema conocida en el tiempo t, mientras que $A_i^p(t)$ es la respuesta del sistema usando la matriz de pesos candidata en el tiempo t.

1.5.3 Algoritmos de aprendizaje híbridos

La meta de los algoritmos de aprendizaje basados en un enfoque híbrido consiste en modificar o actualizar la matriz de pesos causales basados en el conocimiento inicial de los expertos y datos históricos en un proceso de dos fases (Papageorgiou, 2012).

En la literatura se han propuesto pocos algoritmos híbridos. Estos son conformados a partir de una combinación de los algoritmos de aprendizaje hebbianos y los algoritmos de aprendizaje basados en meta-heurísticas. Papageorgiou y Groumpos propusieron el primer algoritmo híbrido en (Papageorgiou and Groumpos, 2005). El funcionamiento de este algoritmo consiste en una combinación del Algoritmo de Evolución Diferencial (DE) y NHL posibilitando así el uso de las potencialidades de la búsqueda global de las estrategias evolutivas y la efectividad de la ley hebbiana. Es válido mencionar que este algoritmo de aprendizaje híbrido fue aplicado satisfactoriamente en problemas reales y sus resultados mostraron que la estrategia híbrida es capaz de entrenar a los MCD de una manera efectiva (Papageorgiou, 2012).

Recientemente Zhu y Zang propusieron un algoritmo híbrido basado en la combinación de RCGA y NHL en (Zhu and Zhang, 2008) posibilitando así que éste combinara el conocimiento del experto y los datos históricos. Los primeros resultados de este algoritmo resultaron prometedores para investigaciones futuras (Papageorgiou, 2012).

1.6 Áreas de Aplicación

Los MCD han sido utilizados y aplicados en muchos campos de investigación como por ejemplo: en la medicina, en las ciencias ambientales y geológicas, en la ingeniería y en la economía, negocios y administración.

En la siguiente tabla mostraremos una selección de aplicaciones por cada una de las áreas mencionadas anteriormente:

Tabla 1.1 Selección de aplicaciones de MCD extraída de (Stach, 2010, Papageorgiou, 2011)

Área de Aplicación	Título	Año
Medicina	• *Decision Making in External Beam Radiation Therapy Based on Fuzzy Cognitive Maps*	2002
	• *An Integrated Two-level Hierarchical System for Decision Making in Radiation Therapy Based on Fuzzy Cognitive Maps*	2003
	• *A Fuzzy Cognitive Map Hierarchical Model for Differential Diagnosis of Dysarthrias and Apraxia of Speech*	2005
	• *Integrating Conventional Science and Aboriginal Perspectives on Diabetes using Fuzzy Cognitive Maps*	2007
	• *Exploring Aboriginal views of Health using*	2008

	Fuzzy Cognitive Maps and Transitive Closure: A Case Study of the Determinants of Diabetes	2008
	• *Brain Tumor Characterization using the Soft Computing Technique of Fuzzy Cognitive Maps*	2008
	• *Fuzzy Cognitive Map based Decision Support System for Thyroid Diagnosis Management*	2009
	• *Fuzzy Cognitive Map Based Approach for Assessing Pulmonary Infections*	2011
	• *Analyzing the performance of fuzzy cognitive maps with non-linear hebbian learning algorithm in predicting autistic disorder*	2011
	• *A new methodology for Decisions in Medical Informatics using Fuzzy Cognitive Maps based on Fuzzy Rule Extraction techniques*	2011
	• *Intuitionistic Fuzzy Cognitive Maps for Medical Decision Making*	
Ciencias ambientales y geológicas	• *Contextual Fuzzy Cognitive Maps for Geographic Information Systems*	1999
	• *Fuzzy Cognitive Mapping as a Tool to Define Management Objectives for Complex Ecosystems*	2002
	• *Analyses of the Environmental Impacts of an Eco-industrial Park using Fuzzy Cognitive Maps*	2003
	• *A Fuzzy Cognitive Mapping Analysis of the Impacts of an Eco-industrial Park*	2004
	• *Ecological Models Based on People's Knowledge: A Multi-step Fuzzy Cognitive Mapping Approach*	2004
	• *Fuzzy Cognitive Maps for Issue Identification in a Water Resources Conflict Resolution System.*	2005
	• *Emergency Management for a Nuclear Power Plant using Fuzzy Cognitive Maps*	2008
	• *Modeling of the High Pressure Core Spray Systems with Fuzzy Cognitive Maps for Operational Transient Analysis in Nuclear Power Reactors*	2009
	• *The Potential of Fuzzy Cognitive Maps for Semi-quantitative Scenario Development, with an Example from Brazil*	2009
	• *Building scenarios with Fuzzy Cognitive Maps: An exploratory study of solar energy*	2011
Ingeniería	• *Fuzzy Cognitive Maps: A Model for Intelligent Supervisory Control Systems*	1999

	• *Dealing with Location Uncertainty in Mobile Networks Using Contextual Fuzzy Cognitive Maps as Spatial Decision Support Systems*	2000
	• *Fuzzy Cognitive Maps for Decision Support in an Intelligent Intrusion Detection System*	2001
	• *Fuzzy Cognitive Map Approach to Web mining Inference Amplification*	2002
	• *Cooperative Cleaning for Distributed Autonomous Robot Systems Using Fuzzy Cognitive Maps*	2003
	• *Modeling Software Development Projects Using Fuzzy Cognitive Maps*	2004
	• *Fuzzy Cognitive Maps for Engineering and Technology Management: What Works in Practice?*	2006
	• *Action Selection in Robots based on Learning Fuzzy Cognitive Map*	2007
	• *Structural Damage Detection using Fuzzy Cognitive Maps and Hebbian Learning*	2011
	• *The Application of Fuzzy Cognitive Map in Soft System Methodology*	2011
	• *Using Fuzzy Cognitive Maps to Support the Analysis of Stakeholders*	2011
Economía, Negocios y Administración.	• *A Fuzzy Cognitive Map-based Stock Market Model: Synthesis, Analysis and Experimental Results*	2001
	• *The Electric Power Market in the United Kingdom: Simulation with Adaptive Intelligent Agents and the use of Fuzzy Cognitive Maps as an Inference Engine*	2004
	• *Fuzzy Cognitive Maps in Business Analysis and Performance-driven Change*	2004
	• *Supply Chain Risk Analysis with Fuzzy Cognitive Maps*	2007
	• *Application of Fuzzy Cognitive Maps for Stock Market Modeling and Forecasting*	2008
	• *Trust Modelling in e-commerce through Fuzzy Cognitive Maps*	2008
	• *An Application of Fuzzy Cognitive Map Based on Active Hebbian Learning Algorithm in Credit Risk Evaluation of Listed Companies*	2009
		2009

	• *An FCM Approach to Better Understanding of Conflicts: a Case of New Technology Development* • *An extension to fuzzy cognitive maps for classification and prediction* • *A case-based reasoning based multi-agent cognitive map inference mechanism: An application to sales opportunity assessment*	2011 2011

Es válido mencionar que las aplicaciones de MCD no se limitan a las áreas tratadas en la tabla anterior. Esta recopilación valida que el campo de estudio de los MCD han ido ganando en relevancia a través de los años y que sus aplicaciones pueden ser muy diversas.

1.7 Conclusiones parciales

Los MCD provienen de la fusión resultante de la teoría de los mapas cognitivos y la lógica difusa. Estos han demostrado ser eficientes en la modelación y simulación de sistemas dinámicos y han sido aplicados en diferentes dominios. Comparados con las redes neuronales y sistemas expertos estos son relativamente más fáciles de usar para la representación de la estructura del conocimiento, y el proceso de inferencia puede ser computado con operaciones en una matriz numérica en vez de reglas explícitas.

La construcción de MCD usualmente es de forma manual a través de expertos o de forma automática y su aprendizaje está enfocado principalmente sobre la estimación de la matriz de pesos causales, debido a que ésta es un aspecto esencial en la fortaleza del modelo. Estos métodos de aprendizaje son divididos según (Papageorgiou, 2012) en tres grupos: algoritmos hebbianos, algoritmos inspirados en meta-heurística y algoritmos híbridos.

Los algoritmos de aprendizaje hebbianos son no supervisados y no son costosos computacionalmente pero requieren de la intervención frecuente del experto en la determinación de parámetros. Los algoritmos inspirados en meta-heurísticas requieren la disponibilidad de datos históricos y son costosos desde el punto de vista computacional debido a la optimización de la función objetivo, pero no requieren frecuentemente de la intervención de los expertos. Los algoritmos de aprendizaje híbridos están enfocados en combinaciones de los dos tipos de aprendizaje mencionados anteriormente, evidentemente son más costosos desde el punto de vista computacional pero han demostrado resultados positivos en su aplicación a problemas reales (Papageorgiou, 2012).

Como se ha mencionado anteriormente la meta de los algoritmos de aprendizaje existentes según (Papageorgiou, 2012) es la estimación de la matriz de pesos causales y hasta donde el autor tiene conocimiento no existen algoritmos de aprendizaje orientados a reducir la dimensión de un MCD o sea estimar el subconjunto mínimo de nodos centrales que caracterizan la modelación en un MCD de manera eficiente, posibilitando así una interpretación más clara del sistema.

CAPÍTULO 2. OPTIMIZACIÓN DE LA TOPOLOGÍA DE UN MCD

En este capítulo se aborda el diseño del algoritmo de aprendizaje supervisado discreto para estimar el conjunto mínimo de nodos centrales en un MCD, optimizando de esta forma la topología que caracteriza el sistema modelado.

El algoritmo de aprendizaje incluye la definición de cinco aspectos fundamentales

1. Codificación de la solución.
2. Diseño de la función objetivo.
3. Diseño de la restricción.
4. Estimación de la información heurística.
5. Selección del optimizador.

2.1 Codificación de la solución

En el diseño del individuo se tuvo en cuenta la presencia de un problema combinatorio con $2^n - 1$ posibles soluciones, donde n denota el número total de conceptos del MCD. Por tanto resulta natural que la representación de una solución sea a través de un vector binario $\vec{y} \in \{0,1\}^n$ donde 0 denota la no pertenencia de una posición (concepto) en el mapa y 1 denota la pertenencia de la posición en la modelación, por ejemplo:

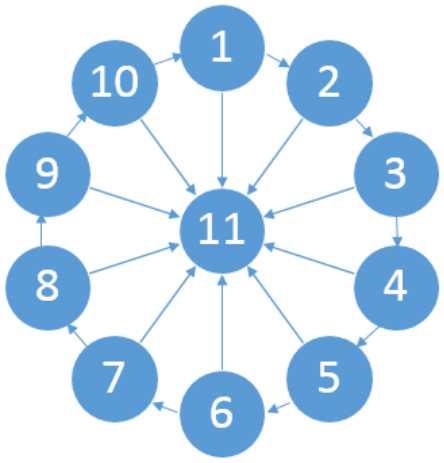


Fig. 2.1 MCD de 11 conceptos.

Como se puede observar en la imagen anterior se tiene una modelación de un MCD de 11 conceptos donde la codificación de una posible solución utilizando este mapa sería la siguiente:

Tabla 2.1 Representación de un individuo

1	1	1	1	1	1	1	1	1	1	0

Donde el concepto cuya posición es la l1 se encuentra codificado por un 0. Por tanto dicho concepto no pertenece a la modelación de esta solución y los conceptos cuyas posiciones correspondan a 1, 2,3, 4, 5, 6, 7, 8, 9,10 se encuentran codificados por 1. Estos conceptos si pertenecen a la modelación (vea la siguiente figura).

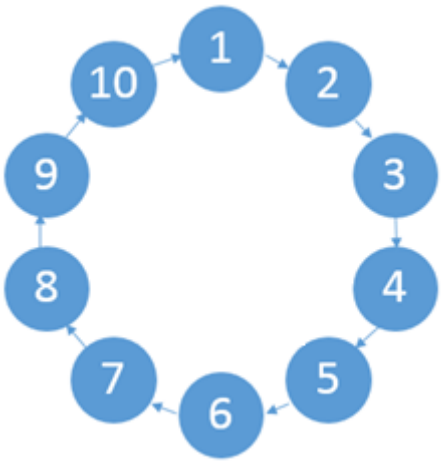


Fig. 2.2 Modelación de la solución.

2.2 Diseño de la función de evaluación

Para conformar la función de evaluación (vea la ecuación 2.2) de un individuo se tuvo en cuenta el balance en cuanto la clasificación del error y reducción de la dimensión del problema.

$$E(\overrightarrow{y},\varphi) = \sum_{i=1}^{|\varphi|} |I(\vec{y},\varphi_i) - R(\varphi_i)| \qquad 2.1$$

$$f(x) = \omega * E(\overrightarrow{y},\varphi) + (1-\omega) * \sum_{i=1}^{n} \overrightarrow{y_i}/_n \qquad 2.2$$

En la expresión anterior $0 < \omega < 1$ es un parámetro que regula la importancia que se le otorga a la clasificación del error, mientras que $(1-\omega)$ regula la reducción en la dimensión. De forma general se recomienda $0.8 \leq \omega < 1$ debido a que se quiere lograr una reducción de la dimensión, pero preservando la exactitud del mapa original. La función $I(\overrightarrow{y},\varphi_i)$ denota la inferencia de la solución para el objeto φ_i de la base de casos y la función $R(\varphi_i)$ representa el valor real de la clase de la instancia φ_i.

2.3 Diseño de la restricción

El proceso de optimización del mapa se logra a través de la conformación de un subconjunto mínimo de conceptos que describen el sistema modelado garantizando que dicho subconjunto preserve la calidad del mapa.

Un componente fundamental para lograr la preservación del error garantizando así la calidad del sistema final es la formulación de restricciones. En esta propuesta se formula una restricción.

Dicha restricción consiste en penalizar (vea ecuación 2.4) a los individuos que no cumplan con una diferencia mínima de clasificación del error del mapa fijada por un umbral $\xi \epsilon$ [0,1] . Para ello se tuvo en cuenta la proporción del error (vea ecuación 2.3) y cómo ésta es reflejada en el intervalo de penalización del individuo.

El control del error es necesario debido a que el modelo resultante de la optimización debe garantizar la calidad del mapa en el grado que el experto lo estime conveniente, por ejemplo: el problema utilizado para validar el algoritmo propuesto es un problema de clasificación de la resistencia de fármacos sobre proteínas del VIH, por lo que resulta evidente que la preservación del error para este problema es sumamente importante, por ello el umbral ξ es fijado a 0.01.

$$e_p = \frac{(e_s - e_o)}{e_s} \tag{2.3}$$

$$f(x) = \begin{cases} f(x) + e_p * \big(1 - f(x)\big), & si\ e_s - e_o > \xi \\ f(x) \qquad\qquad\qquad\quad , & si\ e_s - e_o \leq \xi \end{cases} \tag{2.4}$$

Donde $f(x)$ denota la evaluación de la función objetivo, e_p denota la proporción del error de la solución encontrada respecto al error del mapa original, e_s y e_o denotan el error de la solución encontrada y el error del mapa original respetivamente y $\big(1 - f(x)\big)$ denota el intervalo de penalización del individuo.

Dependiendo del problema se podrán añadir otras restricciones, por ejemplo: en el problema utilizado para validar esta propuesta se añade una referente al tamaño mínimo permitido por una solución, penalizando con el máximo al individuo que no cumpla con esta restricción (en capítulo 3 se aborda dicha restricción).

2.4 Estimación de la información heurística

En esta sección se aborda la estimación de la información heurística sobre el sistema modelado a través de los componentes (conceptos) que lo conforman. Independientemente de si los datos (atributos) que fueron modelados tienen ya prestablecida o no cierta preferencia heurística.

En el caso que no posean preferencia heurística se asume como información heurística de cada componente la establecida en esta sección. Resaltar que éste es el caso de esta propuesta.

En el caso que los atributos modelados posean información heurística ya prestablecida se puede determinar la preferencia heurística de tres formas:

1. Tomar como preferencia heurística de cada componente la información ya prestablecida. Esta opción es recomendable cuando dicha información ya se encuentra probada.
2. Tomar como preferencia heurística de cada componente la propuesta en este trabajo.
3. Realizar un consenso entre las dos opciones mencionadas previamente, por ejemplo: a través del promedio.

En el diseño de la estimación de la información heurística se tuvo en cuenta el rol de los conceptos en la topología del mapa. El nivel de presencia de un concepto C_i en el mapa es representado a través de su grado de activación y la influencia del mismo en el sistema está condicionada por los pesos de sus conexiones tanto las de entrada como la de salida, debido a que las conexiones de entrada determinan el factor de cambio del grado de activación del concepto y las conexiones de salida determinan el factor del cambio del concepto final.

Por ello utilizamos una medida llamada grado de centralidad, la cual nos permite identificar los conceptos centrales determinando su contribución al mapa (vea ecuación 2.2). Ésta es una medida local debido a que está determinada solamente por conexiones directas de entrada y salida del concepto.

$$\eta_i = \frac{1}{2n-2}\left(\sum_{j=1}^{n}|W_{ji}| + \sum_{j=1}^{n}|W_{ij}|\right) \qquad 2.2$$

Donde W_{ji} y W_{ij} denotan el número máximo de conexiones de entrada y salida de un concepto y $\frac{1}{2n-2}$ denota la normalización de esta medida. Es válido destacar que esta medida de estimación de la información heurística asume que no existen las autoconexiones en los conceptos y que todos los conceptos están totalmente conectados por tanto esta es una medida dependiente de las restricciones de la topología del MCD.

2.5 Selección del optimizador

Un aspecto fundamental del algoritmo propuesto es la determinación del método de búsqueda u optimizador debido a que en gran medida la calidad de la optimización dependerá de dicha selección.

Existen diversos métodos de búsqueda en la literatura, específicamente las meta-heurísticas son las opciones viables para esta propuesta producto a la complejidad del espacio de búsqueda de las soluciones.

Resulta evidente que la elección de un optimizador está condicionada por la representación binaria del individuo, cualquier meta-heurística que maneje este tipo de individuo puede ser seleccionada.

En esta propuesta se seleccionó tres variantes de la Optimización basada en Colonias de Hormigas (ACO, por sus siglas en inglés); éstas son:

1. Sistemas de Hormigas(AS, por sus siglas en inglés)
2. Sistema de Colonias de Hormigas(ACS, por sus siglas en inglés)
3. Sistema de Hormigas Max-Min (MMAS, por sus siglas en inglés)

La selección de ACO se realizó debido a que en la literatura consultada (Dorigo and Stützle, 2011) se determinó una gran cantidad de aplicaciones y estudios existentes , por lo que ésta es una de las meta-heurísticas más consolidadas y validadas.

En tal sentido es necesario destacar que no es objetivo de esta propuesta la selección del mejor optimizador, sino demostrar que la propuesta de algoritmo es factible. Sin embargo, es obvio que la potencia del método de búsqueda puede resultar determinante para el desempeño final del algoritmo.

2.5.1 Optimización de Colonia de Hormigas

La inspiración de ACO se basa en el comportamiento de diferentes especies de hormigas reales, y en la forma que estas determinan caminos de trayectoria mínima entre el hormiguero y la fuente de comida, siendo esto especialmente útil para solucionar problemas de optimización combinatoria. En ACO, cada hormiga (agente) tiene en cuenta dos tipos de información cuando iterativamente construye la solución al problema:

1. **Rastros de feromonas artificiales**: Simula la feromona real que depositan las hormigas, la cual se modifica durante el tiempo.
2. **Información heurística**: Denota la preferencia heurística de selección de un estado. Esta información no se modifica durante el progreso del algoritmo.

Esencialmente los rastros de feromona se modelan como valores numéricos y son asociados a componentes de la solución; éstos son adaptados mientras se resuelve una instancia del problema y reflejan la experiencia de búsqueda de la colonia de hormigas artificiales.

2.5.2 Analogía con las hormigas biológicas

Las hormigas son insectos sociales que viven en colonias. Por lo tanto, ellas son capaces de realizar tareas que desde el punto de vista particular no serían posibles. Éstas son capaces de encontrar caminos más cortos del hormiguero a la fuente de comida producto a una sustancia química que es depositada mientras caminan, denominada feromona. De no existir rastro alguno de feromona básicamente la hormiga se mueve de manera aleatoria, pero al existir dicha sustancia, ésta tiende a seguir el rastro.

Experimentalmente algunos investigadores han estudiado el efecto del rastro de feromona en el comportamiento de las hormigas para encontrar formas de cuantificarlo. En (Deneubourg et al., 1990) los investigadores condujeron un experimento llamado experimento de puente binario en *Linepithema humile* (hormigas argentinas) donde la colonia de hormigas estaba conectado a la fuente de alimentos por dos puentes de igual tamaño. Las hormigas podían escoger libremente cual puente cruzar cuando buscaban alimento y regresaban al hormiguero.

En este experimento no existía inicialmente un rastro de feromona en los dos puentes por lo cual las hormigas iniciaban la exploración alrededor del hormiguero y eventualmente cruzaban uno de los dos puentes y alcanzaban el alimento. A partir del desplazamiento de las hormigas desde el hormiguero hasta la fuente de alimentación y viceversa iban depositando la sustancia antes mencionada. Inicialmente cada hormiga escogía un camino aleatorio.

El resultado de este experimento mostró que cada uno de los puentes fue cruzado en un 50% de los casos. En la figura 2.3 se ilustra cómo funciona el rastro de feromona. En este caso es pertinente recordar que inicialmente no existe ningún rastro de feromona en el camino y que la hormiga elige de manera aleatoria el camino a seguir.

Según trascurre el tiempo los caminos más prometedores reciben mayor cantidad de feromonas debido al tamaño del mismo, o sea mientras menos distancia exista entre el hormiguero y la fuente de alimento, la hormiga deposita mayor feromona. En el camino más corto habrá un rastro de feromona superior al resto, por lo que aumenta la probabilidad de que la hormiga privilegie ese camino. Por lo tanto, a medida que transcurra el tiempo y se acumule más sustancia, aumentará la atracción del resto de las hormigas de la colonia.

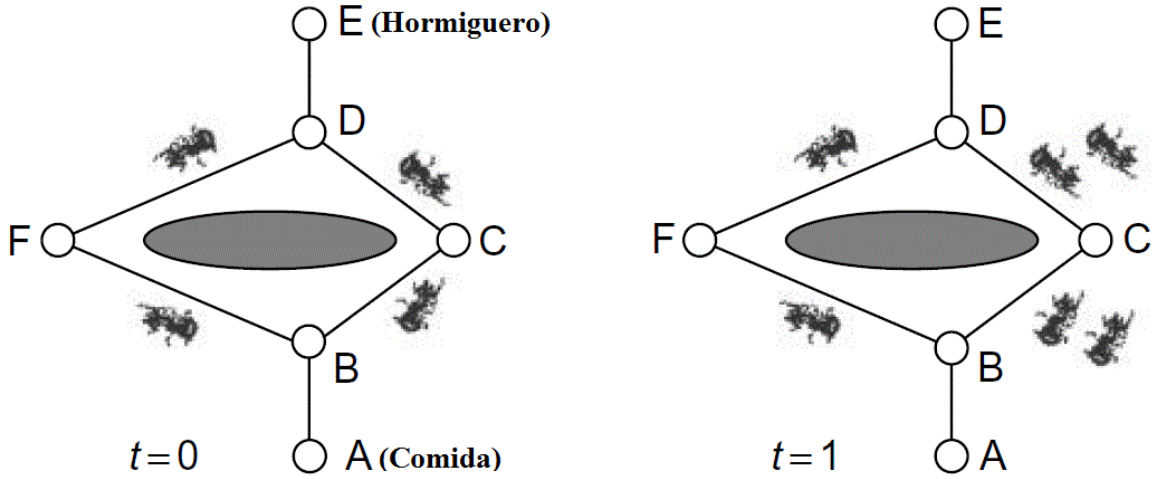


Fig. 2.3 Comportamiento de una colonia de hormigas. Tomada de (Edelkamp and Schroedl, 2011).

Es válido mencionar que esta convergencia también está condicionada por un proceso de evaporación de feromona donde los caminos menos transitados van disminuyendo su rastro de feromona de forma gradual. Sin embargo algunos estudios bilógicos han demostrado que los rastros de feromona son muy persistentes y que dependiendo de la especie de hormiga estos pueden durar horas, e incluso días.

2.5.3 Estructura y funcionamiento de la meta-heurística

El funcionamiento de ACO es el siguiente: m hormigas (agentes computacionales) de la colonia se mueven concurrentemente y de manera asincrónica a través de los posibles estados adyacentes del problema, una posible representación es a través de un grafo. Este movimiento está condicionado por una regla de transición la cual tiene en cuenta los dos tipos de información que dispone la hormiga, la información heurística (información local) y el rastro de feromona (información memorística).

La actualización de los rastros de feromona por las hormigas puede ocurrir en dos momentos:

1. Mientras las hormigas van recorriendo el grafo e incrementalmente construyendo las soluciones (actualización en línea paso a paso del rastro de feromona) en los arcos (conexiones).
2. Cuando las hormigas hayan finalizado la construcción de las soluciones, estas pueden depositar una cantidad de feromona en los arcos que depende de la calidad de la solución encontrada (actualización en línea a posteriori).

En esta propuesta se emplea una modelación diferente debido a que las posiciones del vector no se encuentran relacionadas como ocurre en un grafo (ej. problema del viajero vendedor). Cada posición posee dos estados: pertenencia y no pertenencia. Por tanto los rastros de feromona se representan como

una matriz de τ_{2xn} y la información heurística como una matriz η_{2xn}. La estructura general de un algoritmo ACO es la siguiente:

Procedimiento general de ACO Tomado de (Alonso et al., 2001)
1. Inicialización de parámetros 2. **Mientras**($criterio\ de\ parada\ no\ satisfecho$) 3. **Actividades** 4. **generaciónHormigas** 5. **Mientras**($k < m$) hacer en paralelo 6. nuevaHormiga(k++) 7. **Fin** 8. **Fin** 5. evaporaciónFeromona() 6. accionesDemonio() //opcional 7. **Fin** 8. **Fin**

Procedimiento nuevaHormiga(k) Tomado de (Alonso et al., 2001)
1. inicializaHormiga(k) 2. L=actualizaMemoria() 3. **Mientras**($estadoActual\ != \ estadoObjetivo$) 4. P=probabilidadesTransición(A,L,Ω) 5. siguienteEstado=decisión(P,Ω) 6. moverSiguienteEstado(siguienteEstado) 7. **Si**($actFeromonaLineaPaso\_a\_Paso$) 8. depositarFeromona(arcoVisitado) 9. **Fin** 10. **Fin** 11. **Si**($actFeromonaLineaPosteriori$) 12. **Para (**$cada\ ArcoVisitado$) 13. depositarFeromona(arcoVisitado) 14. **Fin** 15. **Fin** 16. liberarRecursos(k)

Donde A corresponde al conjunto de aristas que conectan el conjunto de componentes, Ω a restricciones del problema, L a la memoria de la hormiga y P a la política a tener en cuenta en la selección del siguiente estado.

Es válido destacar que mediante el término meta-heurística ACO se entiende como el modo genérico de operación. El nombre de algoritmo ACO se utiliza para referirse a una implementación específica de la meta-heurística. En la inicialización de parámetros de forma general se tiene en cuenta los siguientes parámetros:

1. Número de hormigas en la colonia m.

2. El rastro inicial de feromona τ_0 es recomendable que sea un valor pequeño y generalmente es asignado a todas las componentes de solución.
3. Los pesos que definen la importancia que se concede al rastro de feromona α e información heurística β.
4. Tasa de evaporación p, es recomendable sea un valor pequeño.

Por tal motivo, resulta conveniente acentuar que los parámetros mencionados anteriormente pueden diferir en alguna implementación de algoritmo ACO.

El constructor principal la ACO es **Actividades,** éste controla los principales procedimientos de ACO:

1. generaciónHormigas
2. evaporaciónFeromona
3. accionesDemonio

En el procedimiento generaciónHormigas se crean las hormigas y se calculan las probabilidades de selección de un estado y se aplica la política de decisión. Estos pasos están determinados por el algoritmo de ACO que se utilice. Además, el procedimiento actualizaMemoria se encarga de determinar el estado inicial desde que la hormiga parte y almacena el estado correspondiente en la memoria de la hormiga L.

El procedimiento evaporaciónFeromona tiene contrapartida natural como se ha mencionado anteriormente y funciona como mecanismo que trata de evitar el estancamiento de la búsqueda.

El procedimiento accionesDemonio es opcional y no tiene contrapartida natural. Éste brinda la posibilidad de emprender tareas desde una perspectiva global, lo cual no es permisible con las hormigas debido a su enfoque local por ejemplo:

1. Depositar una cantidad extra de feromona en el camino recorrido por la mejor hormiga.
2. Detectar estados de estancamiento y tratar de escapar de ellos.

2.5.4 Sistema de Hormigas

El primer algoritmo de ACO fue AS introducido en (Dorigo et al., 1996). De hecho inicialmente, se presentaron tres variantes distintas que se diferenciaban en la manera en que se actualizaban los rastros de feromona:

1. AS-densidad: aplicaban una actualización en-línea paso a paso de feromona, donde la cantidad de feromona depositada era constante.

2. AS-cantidad: aplicaban una actualización en-línea paso a paso de feromona donde la cantidad de feromona depositada dependía directamente de la preferencia heurística de la transición η_{rs}.
3. AS-ciclo: aplicaba una actualización en línea a posteriori de feromona.

Esta última variante fue la de mejor desempeño y es por tanto la que se conoce como AS en la literatura. El AS se caracteriza por el hecho de que la actualización de feromona se realiza una vez que todas las hormigas han completado sus soluciones, y se lleva a cabo como sigue:

- Todos los rastros de feromona se reducen en un factor constante, implementándose de esta manera la evaporación de feromona.
- Cada hormiga de la colonia deposita una cantidad de feromona que es en función de la calidad de su solución.

Es válido mencionar que inicialmente AS no usa acciones del demonio, aunque estas son fáciles de añadir, por ejemplo: la mejora de las soluciones generadas por la colonia de hormigas a través de un procedimiento de búsqueda local.

El procedimiento de construcción de las soluciones en AS se muestra a continuación. En cada paso de la construcción de una solución, una hormiga *k* selecciona un estado (ir al siguiente nodo) con una probabilidad que se calcula como:

$$p_{rs}^{k} = \begin{cases} \frac{[\tau_{rs}]^{\alpha} \times [\eta_{rs}]^{\beta}}{\sum_{u \, \varepsilon \, N_r^k} [\tau_{rs}]^{\alpha} \times [\eta_{rs}]^{\beta}} \;, & si \; s \in N_k(r) \\ 0 \;, & en \; otro \; caso \end{cases} \qquad 2.5$$

Donde $N_k(r)$ es el vecindario alcanzable desde r que recordar el vecindario de una posición está compuesto por dos estados: presencia o no presencia de un componente, mientras que los parámetros α, β denota la importancia dada a la información de los rastros de feromona e información heurística respectivamente.

En esta formulación α, β ponderan la importancia dada a la información de los rastros de feromona e información heurística y su función es la siguiente:

- Si $\alpha = 0$, aquellos nodos con una preferencia heurística mejor tienen una mayor probabilidad de ser seleccionados, haciendo el algoritmo muy similar a un algoritmo voraz probabilístico clásico.
- Si $\beta = 0$, sólo se tienen en cuenta los rastros de feromona lo cual puede causar estancamiento. Además los rastros de feromona asociados a una solución pueden ser ligeramente superiores que el resto causando así convergencia prematura a óptimos locales.

Por tanto es preciso establecer una adecuada proporción entre la información heurística y la información de los rastros de feromonas. Es recomendable dar mayor importancia a los rastros de feromona debido a que estos representan la experiencia de búsqueda aunque en algunos casos la información heurística tiene preferencia. Básicamente la importancia de estos parámetros está en dependencia del problema a tratar. A continuación se muestra el procedimiento nuevaHormiga para este algoritmo de ACO en particular:

Procedimiento nuevaHormiga(k) tomado de (Alonso et al., 2001)	
1.	r=estadoInicial()
2.	S_k=r
3.	L_k=r
4.	**Mientras**($estadoActual != estadoObjetivo$)
5.	**Para**($cada\ s \epsilon N_k(r)$)
6.	$P^k_{rs}=\begin{cases} \frac{[\tau_{rs}]^\alpha \times [\eta_{rs}]^\beta}{\sum_{u\ \varepsilon\ N_r^k} [\tau_{rs}]^\alpha \times [\eta_{rs}]^\beta}\ ,\ si\ s \in N_k(r) \\ 0\ ,\ en\ otro\ caso \end{cases}$
7.	**Fin**
8.	siguienteEstado=decisión(P,N_k(r))
9.	r=siguienteEstado
10.	S_k=< S_k ,r>
11.	---
12.	L_k= L_k ∪ r
13.	**Fin**
	/* Se evapora feromona en cada arco a_{rs} * $\tau_{rs} = (1-p) \times \tau_{rs}$ */
14.	**Para (**$cadaArcoVisitado\ a_{rs}$)
15.	$\tau_{rs} = \tau_{rs} + f(C(S_k))$
16.	**Fin**
17.	liberarRecursos(k)

La línea 11 se mantiene vacía para resaltar que en este algoritmo no existe una actualización de feromona en línea paso a paso.

Entre el paso 4 y paso 13 una hormiga k construye la solución, en esta propuesta el criterio de parada es hasta que la hormiga seleccione los estados (pertenencia o no pertenencia) de las n posiciones. O sea, para cada posición de calculan dos probabilidades, pertenencia p_1 y no pertenencia p_0 y en el procedimiento decisión se mapean las dos probabilidades en una ruleta donde cada probabilidad es representada o sea la ruleta tiene dos posiciones, este comportamiento es reflejado en el siguiente seudocódigo:

Seudocódigo Ruleta	
1.	ruleta[0]=p_0
2.	ruleta[1]=ruleta[0]+p_1

```
3.  rand= Random*ruleta[1]
4.  Si(rand ≤ ruleta [0])
5.     estadoPosición= noPertenencia
6.  sino
7.     estadoPosición= pertenencia
8.  Fin
```

Recordar que en AS el procedimiento de evaporación de feromona se ejecuta luego de que todas las hormigas han completado sus soluciones, este procedimiento es ejecutado por el demonio, donde $p \epsilon (0,1]$ denota la tasa de evaporación (vea siguiente ecuación):

$$\tau_{rs} = (1 - p) * \tau_{rs} \quad 2.6$$

En los pasos del 14 al 16 se realiza la actualización de la feromona, donde se tiene en cuenta la calidad de la solución en la cantidad de feromona depositada (vea siguiente ecuación):

$$\tau_{rs} = \tau_{rs} + f(C(S_k)) \quad 2.7$$

Donde $C(S_k)$ denota la calidad de la solución o sea su evaluación y f denota como esta influye en la actualización de feromona, que al ser un problema de minimización se diseñó de la siguiente manera:

$$f(x) = \frac{1}{x} \quad 2.8$$

Es válido mencionar que los autores de AS también propusieron una versión extendida del algoritmo llamada AS elitista (Dorigo et al., 1996), donde ésta generalmente mejoraba los resultados obtenidos.

Esta extensión consistía en la deposición adicional de feromona de la mejor solución global por el demonio en base a la calidad de la misma, además de incrementar dicha deposición en un factor e, este comportamiento es descrito en la siguiente ecuación:

$$\tau_{rs} = \tau_{rs} + e * f\left(C(S_{gb})\right) \quad \forall a_{rs} \epsilon S_{gb} \quad 2.9$$

Donde e denota el número de hormigas elitistas y S_{gb} denota la mejor solución global.

2.5.5 Sistema de Colonias de Hormigas

ACS (Dorigo and Gambardella, 1997) es uno de los primeros sucesores de AS que introduce mejoras significativas incrementando la importancia de la explotación de la información recolectada por las hormigas anteriores sobre la exploración del espacio de búsqueda. Esto se logra a través de dos mecanismos: en primer lugar se utiliza una estrategia elitista para actualizar el rastro de feromona y las

hormigas seleccionan el estado de una posición usando una regla llamada regla proporcional pseudo-aleatoria (Dorigo and Gambardella, 1997).

Las principales modificaciones con respecto a AS son las siguientes (Alonso et al., 2001) :

1. El ACS usa una regla de transición distinta, denominada regla proporcional pseudo-aleatoria. Sea k una hormiga situada en el nodo r, $q_0 \epsilon [0,1]$ un parámetro y q un valor aleatorio en $[0,1]$, el estado de una posición (el siguiente nodo) se elige mediante la siguiente distribución de probabilidad:

$$p_{rs}^{k} = \begin{cases} \begin{cases} 1\ , si\ s = \arg\max_{u\ \epsilon\ N_k(r)} \left\{ \tau_{ru}\ \times\ \eta_{ru}^{\beta} \right\} \\ 0\ , \qquad\qquad en\ otro\ caso \end{cases} & ,\quad si\ q \le q_0 \\ \begin{cases} \frac{[\tau_{rs}]^{\alpha} \times [\eta_{rs}]^{\beta}}{\sum_{u\ \epsilon\ N_r^k} [\tau_{rs}]^{\alpha} \times [\eta_{rs}]^{\beta}}\ , & si\ s\ \epsilon\ N_k(r) \\ 0 \qquad\qquad , & en\ otro\ caso \end{cases} & ,\quad si\ q > q_0 \end{cases} \qquad 2.10$$

Si el valor de q_0 es cercano a uno, como sucede en la mayoría de las aplicaciones de ACS, la explotación del conocimiento disponible es privilegiada o sea en la mayoría de los casos se elige la probabilidad de un estado basado en la información heurística y los rastros de feromona. Obviamente si $q > q_0$ se aplica la misma regla de probabilidad que en AS.

2. Se realiza una actualización de feromona fuera de línea de los rastros. Para llevarla a cabo, el ACS sólo considera una hormiga concreta, aquella que generó la mejor solución global, S_{gb} (aunque en algunos trabajos iniciales se consideraba también una actualización basada en la mejor hormiga de la iteración (Dorigo and Gambardella, 1997), en ACO casi siempre se aplica la actualización por medio de la mejor global).

Al igual que en AS a la actualización de feromona la precede un procedimiento de evaporación de feromona con la diferencia que no todas las conexiones sufren la evaporación de feromona solamente las conexiones que recorrió la mejor solución global:

$$\tau_{rs}\ = (1 - p)\ *\ \tau_{rs}\ \forall a_{rs} \epsilon\ S_{gb} \qquad 2.11$$

Luego el demonio realiza la deposición de feromona utilizando la mejor solución global:

$$\tau_{rs}\ =\ \tau_{rs}\ + p * f\left(C\left(S_{gb}\right)\right)\ \ \forall a_{rs} \epsilon\ S_{gb} \qquad 2.12$$

3. Finalmente ACS difiere de AS en que las hormigas mientras construyen las soluciones realizan una actualización en línea paso a paso de los rastros de feromona (similar a como se realizaba en AS-

densidad y AS-cantidad), lo cual favorece la generación de soluciones distintas a las ya encontradas (vea siguiente ecuación).

$$\tau_{rs} = (1-\varphi) * \tau_{rs} + \varphi * \tau_0 \qquad 2.13$$

Como puede observase la actualización en línea paso a paso de los rastros de feromona incluye la evaporación de feromona y la deposición de la misma; $\varphi \epsilon (0,1]$ es un segundo parámetro de decremento de feromona. A continuación mostramos el comportamiento de este algoritmo ACO a través de los procedimientos nuevaHormiga y accionesDemonio.

Procedimiento nuevaHormiga(k) tomado de (Alonso et al., 2001)
1. r=estadoInicial() 2. S_k=r 3. L_k=r 4. **Mientras**($estadoActual != estadoObjetivo$) 5. **Para**($cada\ s \epsilon N_k(r)$) 6. $b_{rs}=\tau_{rs} * [\eta_{rs}]^{\beta}$ 7. **Fin** 8. q=random[0,1] 9. **Si**($q \leq q_0$) 10. siguienteEstado=max(b_{rs}, N_k(r)) 11. **sino** 12. **Para**($cada\ s \epsilon N_k(r)$) 13. $P^k_{rs}=\begin{cases} \frac{[\tau_{rs}]^{\alpha} \times [\eta_{rs}]^{\beta}}{\sum_{u\ \varepsilon\ N_r^k} [\tau_{rs}]^{\alpha} \times [\eta_{rs}]^{\beta}}, & si\ s \epsilon N_k(r) \\ 0, & en\ otro\ caso \end{cases}$ 14. **Fin** 15. siguienteEstado=decisión(P,N_k(r)) 16. **Fin** 17. r=siguienteEstado 18. S_k=< S_k ,r> 19. $\tau_{rs} = (1-\varphi) \times \tau_{rs} + \varphi \times \tau_0$ 20. L_k= $L_k \cup$ r 21. **Fin** 22. liberarRecursos(k)

Procedimiento accionesDemonio tomado de (Alonso et al., 2001)
1. **Para**($cada\ S_k$) 2. BúsquedaLocal(S_k) // opcional 3. **Fin** 4. S_b=MejorSolución(S_k) 5. **Si**($S_b\ es\_mejor\ S_{gb}$) 6. S_{gb}= S_b 3. **Fin** 4. **Para**($cada\ arista\ a_{rs} \epsilon S_{gb}$) /*****Se evapora feromona en cada arco a_{rs} de la mejor * solución global**:** $\tau_{rs} = (1-p) * \tau_{rs}$ *****/

5. $\tau_{rs} = \tau_{rs} + p * f\left(C(S_{gb})\right)$
6. **Fin**
7. LiberarRecursos(k)

2.5.6 Sistema de Hormigas Max-Min

MMAS (Stützle and Hoos, 1996) es una de las extensiones de AS que mejor desempeño muestran. Extiende a AS en los siguientes aspectos (Alonso et al., 2001):

1. Se aplica una actualización fuera de línea de los rastros feromona, utilizando para ello la mejor solución global (vea siguiente ecuación) y aplicando previamente una evaporación de feromona a todos los rastros de feromona (vea ecuación 2.11). Puede apreciarse que esta forma de actualización es similar a la utilizada en ACS.

$$\tau_{rs} = \tau_{rs} + f\left(C(S_{gb})\right) \quad \forall a_{rs} \epsilon S_{gb} \qquad 2.14$$

Aclarar que en la selección de la mejor hormiga para actualizar los rastros de feromona se ha demostrado experimentalmente (Stützle, 1999, Stützle and Hoos, 2000) que el mejor rendimiento se obtiene aumentando la frecuencia de selección de la mejor hormiga global sobre la mejor hormiga de la iteración. Este enfoque es utilizado en esta propuesta.

2. En MMAS se introduce límites inferiores y superiores a los valores de los rastros de feromona, además de una diferente inicialización de dichos valores. Por tanto, los rastros de feromona están limitados al rango $[\tau_{min}, \tau_{max}]$. Existen varias heurísticas para calcular los límites de feromona, en esta propuesta para el cálculo de τ_{max} se utilizó la siguiente ecuación. La selección de esta heurística para el cálculo de τ_{max} estuvo condicionada por el hecho de que el nivel máximo de feromona se alcanza cuando S^* es la solución óptima debido al procedimiento de evaporación de feromona.

$$\tau_{max} = \frac{1}{(p \times C(S^*)} \qquad 2.15$$

Donde S^* puede ser la mejor solución global o la mejor solución de la iteración. Para el cálculo de τ_{min} solo basta que este sea un valor constante menor que τ_{max} en esta propuesta utilizamos la siguiente ecuación:

$$\tau_{min} = \frac{\tau_{max}}{n} \qquad 2.16$$

3. Los rastros iniciales de feromona en MMAS se inicializan de manera diferente a como se inicializaban en AS y ACS. Estos se inicializan con el límite superior de feromona permitido τ_{max}, para el cálculo inicial de este se puede obtener una solución utilizando una heurística voraz y sustituirla en S^*. En esta propuesta en el cálculo inicial de τ_{max} S^* se obtiene de manera aleatoria aumentando así la eficiencia del algoritmo.

A continuación se muestra el comportamiento de este algoritmo ACO a través del procedimiento accionesDemonio().

Procedimiento accionesDemonio Tomada de (Alonso et al., 2001)	
1.	**Para**($cada\ S_k$)
2.	**b**úsquedaLocal(S_k)
3.	**Fin**
4.	S_b=mejorSolución(S_k)
5.	**Si**($S_b\ es\_mejor\ S_{gb}$)
6.	S_{gb}= S_b
7.	**Fin**
8.	S_m=probabilidad(S_{gb},S_b)
9.	**Para**($cada\ arista\ a_{rs} \epsilon\ S_m$)
10.	$\tau_{rs} = \tau_{rs} + f(C(S_m))$
11.	**Si**($\tau_{rs} < \tau_{min}$)
12.	$\tau_{rs} = \tau_{min}$
13.	**Fin**
14.	**Fin**
15.	**Si**(condiciónEstancamiento)
16.	**Para**($cada\ arista\ a_{rs}$)
17.	$\tau_{rs} = \tau_{max}$
18.	**Fin**
19.	**Fin**
20.	liberarRecursos(k)

2.6 Búsqueda Local

La búsqueda local es una de las formas más naturales de intentar solucionar un problema de optimización combinatoria. Dada la una solución, la idea es tratar de mejorar su valor a partir cambios locales en los componentes de la solución. Algunos de estos cambios pueden ser intercambiar componentes (cambiar el orden), eliminarlos o añadirlos.

Luego de efectuar alguna de las opciones anteriores si se encuentra una solución mejor que la solución inicial, entonces se inicia nuevamente el procedimiento con la nueva solución hasta que no se obtenga ninguna mejora.

Independientemente del optimizador seleccionado en esta propuesta se aplica un procedimiento de refinamiento de las soluciones obtenidas por los individuos aplicando búsqueda local. Para cada individuo decidimos sustituir o no el estado de sus posición por el estado del mejor individuo global utilizando para ello una regla de generación hacia el extremo global tomada del algoritmo Optimización basada en mallas dinámicas (Bello and Puris, 2008). El siguiente seudocódigo describe lo anteriormente expuesto:

Procedimiento mejoraIndividuo(k)
1. **Si**($Individuo_k$ no_es_mejor $Individuo_{gb}$))
2. **Hacer**
3. $r = 1 - 0.5 * \frac{\text{Individuo}_{gb}}{\text{Individuo}_k}$
4. **Para** ($i < n$)
5. **Si**($Random < r$)
6. Individuo $_{lb}$[i]= Individuo$_{gb}$[i]
7. **Sino**
8. Individuo $_{lb}$[i]= Individuo$_{k}$[i]
9. **Fin**
10. **Fin**
11. **Si**($Individuo_{lb}$ es_mejor $Individuo_k$)
12. Individuo $_k$=Individuo $_{lb}$
13. **Fin**
14. **Si**($Individuo_{lb}$ es_mejor $Individuo_{gb}$)
15. terminarProcedimiento()
16. **Fin**
17. **Mientras**($mejoro\ Individuo_k$)
18. **Fin**

Donde Individuo $_k$ denota la solución alcanzada por la individuo k, $Individuo_{gb}$ denota la solución alcanzada por el mejor individuo global y Individuo $_{lb}$denota la solución intermedia generada utilizando el Individuo $_k$ y la $Individuo_{gb}$.

2.7 Conclusiones parciales

En el algoritmo de aprendizaje propuesto para la optimización de MCD se determinó que la codificación más adecuada del individuo es la representación binaria. Además se precisó el balance tenido en cuenta en la definición de los parámetros de la función de evaluación objetivo.

Se resaltó la necesidad de la definición de restricciones, específicamente la relacionada con el control del error y cómo ésta es regulada por el experto. Seguidamente se detalló las distintas maneras de cómo determinar la preferencia heurística de un concepto, proponiendo a la vez una variante para estimar la preferencia heurística del mismo sobre el modelo.

Es pertinente recalcar que en la selección del optimizador no se tuvo en cuenta la elección del mejor método de búsqueda debido a que no es objetivo de esta propuesta. No obstante la potencia de éste resulta determinante en el desempeño del algoritmo.

El algoritmo de refinamiento de las soluciones obtenidas es independiente del optimizador utilizado y se aplica luego de determinada la población de individuos en cada iteración del optimizador.

CAPÍTULO 3. EVALUACIÓN DEL ALGORITMO PROPUESTO

En este capítulo se aborda la validación del algoritmo de aprendizaje supervisado discreto propuesto para estimar el conjunto mínimo de nodos centrales de un MCD. Esto se efectúa a través de un problema real de clasificación de la resistencia de once mapas de fármacos antivirales de dos proteínas del VIH la proteasa y la transcriptasa inversa.

3.1 Modelación de proteínas del VIH usando Mapas Cognitivos Difusos

En las siguientes sub-secciones se describirá el problema real con el cual será validada la propuesta de algoritmo, además de mostrar la construcción del mapa cognitivo difuso para dicho problema y la estimación e interpretación de las causalidades.

3.1.1 Problema biológico

El Virus de la Inmunodeficiencia Humana (VIH) es un lentivirus de la familia *Retroviridae*. Este se divide en dos tipos: VIH-1, el más virulento e infeccioso causando anualmente millones de muertes y el VIH-2, menos contagioso, y por ello confinado casi exclusivamente en países de África Occidental. Desde el punto de vista biológico, el genoma del VIH es una cadena ARN codificada por varias proteínas que son esenciales en su ciclo de vida. Por ejemplo, la proteína proteasa es requerida por el VIH para el ensamblaje final de los viriones, en el proceso de replicación viral, si la actividad de la proteasa fuera inhibida, la replicación viral del VIH sería considerablemente reducida.

Otra proteína requerida para la replicación del virus es la transcriptasa inversa, la cual cataliza el proceso de transcripción inversa transformando ARN en ADN, este ADN es incorporado dentro del genoma del huésped y "programa" la célula del huésped para producir nuevas partículas virales que son liberadas, infectando a nuevas células y completando el ciclo de vida del virus (Kierczak, 2009).

Diversos fármacos antivirales han sido diseñados para fijarse en el centro activo de estas proteínas e inhibir sus funciones, reduciendo la replicación viral y prolongando la vida de los pacientes infectados por el VIH. Usualmente los fármacos son exitosamente combinados en un tratamiento antiviral activo. Sin embargo, debido a la capacidad de mutación y las características de adaptación dinámica, el virus es capaz de desarrollar resistencia a los fármacos existentes y eventualmente causar el fallo del tratamiento. Por este motivo resulta relevante el estudio de los mecanismos de resistencia de este virus, con el objetivo de diseñar de tratamientos efectivos usando fármacos existentes.

El test de resistencia de una mutación de una proteína puede efectuarse por dos métodos: el test de resistencia fenotípica, que mide la actividad viral en la presencia de una concentración de un fármaco; y

el test de resistencia genotípica que secuencia los genes del virus y realiza un estudio de las mutaciones asociadas con la resistencia. Los ensayos fenotípicos son muy exactos pero también costosos, mientras que los ensayos genotípicos son baratos en cuanto a tiempo y esfuerzo pero solo proveen información indirecta de la resistencia, brindando un conocimiento limitado del comportamiento del VIH (Beerenwinkel, 2002).

En los últimos años los resultados de ambos experimentos han sido combinados en pares fenotipo-genotipo, para así obtener información fenotípica más relevante y consistente del genotipo, utilizando métodos biológicos, matemáticos o computacionales.

3.1.2 Construcción del MCD

La proteasa y transcriptasa inversa son modeladas como un sistema dinámico utilizando la teoría de los MCD. Este modelo permite predecir la resistencia del fármaco dada una mutación en la secuencia de la proteasa e interpretar las relaciones causales entre los aminoácidos y su influencia en la resistencia.

La secuencia de la proteína proteasa está definida por 99 aminoácidos, mientras que la transcriptasa inversa tiene 240 aminoácidos donde cada uno es caracterizado por la energía de contacto. La energía de contacto (Miyazawa, 1999) de un aminoácido es un descriptor numérico que corresponde con la estructura tridimensional de la proteína. Este descriptor se calcula estadísticamente a partir de diversas secuencias, y su papel consiste en caracterizar el funcionamiento de los aminoácidos.

Para modelar las proteínas del VIH cada aminoácido (posición de la secuencia) es tomado como un concepto del mapa, y otro concepto es definido para la resistencia. Además, se establecen las relaciones causales entre todos los aminoácidos y cada uno de ellos con la resistencia. Esta topología se apoya en la existencia de relaciones entre posiciones no necesariamente adyacentes de la secuencia debido a la estructura tridimensional de la proteína, donde un cambio en la energía de contacto de una posición específica (considerada como una mutación del aminoácido) puede ser relevante desde el punto de vista de la resistencia. La siguiente figura ilustra la topología descrita.

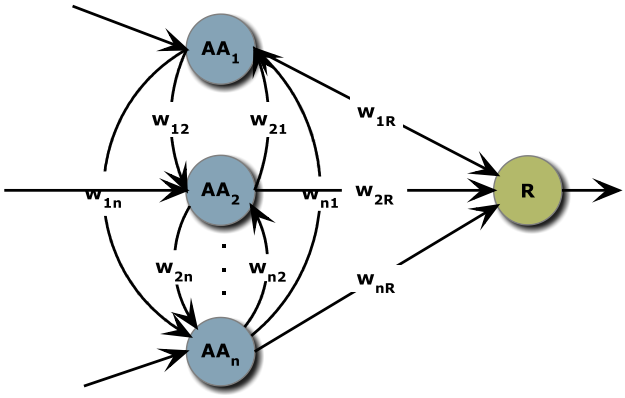


Fig. 3.1. Esquema general de modelación de las proteínas del VIH con MCD.

La predicción del concepto resistencia para cada fármaco significa resolver el problema de clasificación de la secuencia. El valor de activación de cada atributo es tomado como la normalización de la energía de contacto respecto a la máxima energía de contacto. Además, los conceptos descriptores (aminoácidos) usan una función sigmoidal para mantener el nivel de activación en el rango [0.1].

El concepto *resistencia* es representado como un concepto binario (1- resistente, 0-suceptible) definido por un fármaco específico. Las relaciones causales entre los conceptos son definidas por una matriz de pesos causales $W_{nx(n+1)}$ que inicialmente sus valores son asignados aleatoriamente en el rango $[-1,1]$. Esta matriz define, para cada aminoácido AA_i, la relación causal con otras posiciones (AA_j) y además caracteriza la relación causal entre cada aminoácido AA_i y el concepto resistencia, denotado como R. En este modelo se asume que las autoconexiones de un concepto no son permitidas.

3.1.3 Estimación de las causalidades

Como se mencionaba anteriormente, la estimación de las causalidades del mapa se realiza de manera aleatoria lo cual trae consigo que el mapa no esté debidamente entrenado. Por lo cual se aplica un algoritmo de aprendizaje basado en meta-heurísticas enfocado a estimar la matriz de pesos causales a través de datos históricos. La meta-heurística utilizada para estimar la matriz de pesos causales fue una variante de Optimización basada en Bandadas de Partículas (PSO-RSVN)(Nápoles et al., 2012b, Nápoles et al., 2012a) la cual es capaz de detectar estados potenciales de estancamiento o convergencia prematura.

En el proceso de estimación de la matriz de pesos causales, cada agente de la bandada denota una posible matriz, estos agentes son tratados como vectores. En el proceso de optimización, el espacio de búsqueda es explorado tratando de encontrar la mejor configuración que garantice la convergencia. Este método de aprendizaje mejora la calidad de los modelos de MCD minimizando la siguiente función objetivo:

$$f(\vec{x}, \varphi) = \sum_{i=1}^{|\varphi|} |I(\vec{x}, \varphi_i) - R(\varphi_i)| \quad (3.1)$$

En la ecuación anterior φ representa una lista de mutaciones de la proteasa que describen el comportamiento biológico de la proteína y φ_i nos da respuesta de la mutación en términos de susceptibilidad y resistencia en contra de cierto fármaco. Específicamente φ denota la base de conocimiento usada en el proceso de aprendizaje, la cual permite estimar la causalidad sobre el sistema modelado. I es una función binaria que calcula la inferencia del mapa para la i-ésima mutación usando la matriz de pesos $\vec{x}$. R es otra función binaria usada para computar la resistencia del fármaco para la mutación φ_i de la base de conocimientos.

La función obtiene una evaluación óptima cuando el proceso de inferencia y el criterio de resistencia almacenado en la base de conocimiento son idénticos, para todas las mutaciones, lo que significa que el error de simulación respecto a la información biológica no es relevante para poder predecir de forma eficiente la resistencia del fármaco del VIH.

3.1.4 Interpretación de las causalidades

Esta modelación fue concebida para obtener patrones en las relaciones causales entre cada aminoácido y el concepto resistencia (llamadas relaciones directas). Para interpretar las causalidades del mapa resultante, luego de aplicar el algoritmo propuesto, solamente se toma una porción del mapa, la cual corresponde a las relaciones directas. Esta porción está relacionada con las relaciones causales que explican el efecto de una determinada mutación en la resistencia. A continuación se muestra una tabla con patrones descubiertos en un mapa de ejemplo; donde un valor positivo significa que existe una causalidad positiva; un valor negativo significa que existe una causalidad negativa y un valor neutro significa que no es significante la causalidad entre el aminoácido y la resistencia.

Tabla 3.1 Influencias causales de los aminoácidos sobre la resistencia para un fármaco
Tomada de (Nápoles et al., 2014).

AA_{10}	AA_{20}	AA_{36}	AA_{46}	AA_{48}	AA_{50}	AA_{53}	AA_{54}
0.474	-0.414	-0.074	-0.167	-0.411	-0.311	-0.571	0.218
AA_{63}	AA_{71}	AA_{73}	AA_{82}	AA_{84}	AA_{90}	AA_{93}	
0.105	-0.141	-0.114	-0.100	0.384	0.605	0.004	

Analizando los patrones causales descritos en la tabla anterior, se puede arribar a las siguientes conclusiones (Nápoles et al., 2014) :

Para las relaciones directas AA_{10}, AA_{54}, AA_{63}, AA_{84} y AA_{90} existe una causalidad positiva; lo cual significa que cuando una mutación tome lugar en alguno de estos aminoácidos, el valor de la resistencia será proporcionalmente afectado.

Para las relaciones directas AA_{20}, AA_{46}, AA_{48}, AA_{50}, AA_{53}, AA_{71}, AA_{73} y AA_{82} existe una causalidad negativa; lo que significa que cuando una mutación en estas posiciones ocurra, el valor de la resistencia será afectado inversamente proporcional.

Para las relaciones directas AA_{36} y AA_{93} no hay influencias causales significativas sobre el concepto resistencia. Sin embargo, una mutación en alguno de estos dos aminoácidos puede ser relevante para todo el mapa debido a las interacciones de este concepto con el resto. Los patrones de mutaciones descubiertos caracterizan el efecto de una mutación en el valor de la resistencia, proporcionando a los expertos mayor compresión de este complejo y dinámico sistema biológico.

3.2 Configuración de los experimentos

Los experimentos se realizaron en once mapas de inhibidores de la proteasa y la transcriptasa inversa. Específicamente siete mapas de la proteasa: Amprenavir (APV), Atazanavir (ATV), Indinavir (IDV), Lopinavir (LPV), Nelfinavir (NFV), Ritonavir (RTV) y Saquinavir (SQV) y cuatro mapas de la transcriptasa inversa: Didanosine (DDI), Delavirdine (DLV), Efavirenz (EFV) y Nevirapine (NVP).

En las siguientes sub-secciones se describen las bases de conocimiento utilizadas en la modelación de los mapas, una restricción propia del problema tratado y la configuración de parámetros utilizada en la configuración de los tres algoritmos ACO seleccionados como optimizadores.

3.2.1 Descripción de las bases de conocimiento

Las bases de conocimiento correspondientes a los once fármacos antivirales seleccionados fueron construidas utilizando el formato (*Attribute-Relation File Format, ARFF*), el cual es el mismo utilizado por la plataforma de aprendizaje automático WEKA[1].En la siguiente tabla se muestra la cantidad de atributos seleccionados por base:

[1] WEKA es una plataforma de software para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato, Nueva Zelanda.

Tabla 3.2 Cantidad de atributos seleccionados por base.

APV	ATV	IDV	LPV	NFV	RTV	SQV	DDI
51	59	48	51	40	44	48	38
DLV	**EFV**	**NVP**					
44	41	45					

A continuación se analiza la significación de los datos de la base de conocimiento. Estos datos fueron normalizados a través de la división de los mismos por la máxima energía de contacto y el valor de la clase también fue normalizado mediante la división del mismo por el mayor valor de la clase (maxValue) y si este número es menor de 3.5/maxValue entonces la clase adoptaba el valor 0 sino 1. La constante 3.5 utilizada para la determinación de si una instancia es resistente o susceptible parte de la literatura consultada.

La secuencia de números mostrada a continuación en cada fila corresponde a una mutación de la proteína. Cada posición está caracterizada por la energía de contacto de cada uno de los aminoácidos y el número al final de cada fila muestra la influencia en la resistencia (1- resistente, 0-suceptible).

@data

8.38236,2.995464,8.783541,3.580824,7.182885,8.38236,7.217112,8.011692,4.128588,7.182885,3.672966,7.182885,8.011692,3.909708,4.455882,7.182885,3.786624,7.217112,8.783541,1

8.011692,3.999294,8.783541,3.580824,7.182885,8.38236,8.011692,8.011692,7.182885,3.909708,3.672966,7.182885,7.182885,4.128588,4.883514,8.011692,3.786624,8.783541,8.783541,1

8.011692,2.995464,8.783541,3.580824,7.182885,8.783541,7.217112,7.217112,7.182885,7.182885,3.672966,7.182885,8.011692,4.128588,4.883514,7.182885,3.786624,8.783541,8.783541,1

8.011692,2.995464,8.011692,3.580824,7.182885,8.783541,7.217112,8.783541,4.128588,7.182885,3.672966,7.182885,8.011692,4.128588,4.883514,7.182885,3.786624,8.783541,8.011692,1

3.2.2 Una nueva restricción del problema biológico

El problema utilizado para validar el algoritmo propuesto trae consigo la siguiente restricción: el tamaño mínimo permisible de conceptos de una solución es diez, esto es debido a que en la literatura consultada el centro activo de las proteínas del VIH o sea el núcleo no tiene menos de 10 aminoácidos (Nápoles and Grau, 2013). Por tanto los individuos cuyas soluciones no cumplan con la restricción antes expuesta serán penalizados con el máximo valor que puede tomar la función de evaluación (vea siguiente ecuación).

$$f(x) = \begin{cases} 1 & , \quad si \ \overrightarrow{y}_i < 10 \\ f(x), & si \ \overrightarrow{y}_i \geq 10 \end{cases} \qquad 3.2$$

3.2.4 Configuración de los parámetros

La configuración de los parámetros utilizada en los tres algoritmos ACO fue la siguiente:

- El criterio de parada fue el número de evaluaciones de la función objetivo, este fue fijado a 10000. El tamaño del espacio de búsqueda promedio teniendo en cuenta los once mapas es de 2^{46} podemos apreciar que la cantidad de soluciones evaluadas representa un porciento ínfimo del espacio de búsqueda.
- El número de hormigas m fue fijado a diez
- La tasa de evaporación p fue fijada a 0.1.
- La importancia dada a la información heurística β fue de 2 y a los rastros de feromona α fue de 3.
- La importancia w dada a la preservación del error fue de 0.95 por tanto a la reducción de la dimensión del mapa fue fijado a 0.05.
- En ACS se tiene un segundo parámetro de decremento de feromona φ y este se fijó a 0.1. Este valor fue determinado durante la experimentación.
- En ACS involucra el parámetro q_0 de la regla probabilística a utilizar. Éste fue determinado durante la experimentación y fue fijado a 0.8.

En cuanto a los rastros de feromonas iniciales:

- En AS el cálculo de los rastros iniciales de feromona τ_0 se efectuó de la siguiente manera: $\frac{p}{2}$ o sea 0.05.Recordar que p denota la tasa de evaporación.
- En ACS τ_0 se fijó a : 0.5. Este valor fue determinado por experimentación.
- En MMAS el cálculo de los rastros iniciales de feromona se efectuó de la siguiente manera: $\tau_0 = \frac{1}{(p \times C(S^*)}$. Recordar que esta inicialización corresponde con la definición de τ_{max} y $\tau_{min} = \frac{\tau_{max}}{n}$ .Una explicación más detallada de estas heurísticas se encuentra en el capítulo 2.

3.3 Análisis estadístico

En esta sección se analizan los resultados obtenidos luego de efectuar para el algoritmo propuesto utilizando como optimizador cada uno de los algoritmos de ACO analizados en el capítulo 2, diez corridas para cada uno de los once mapas mencionados anteriormente y los resultados mostrados son el promedio de dichas corridas. Para cada corrida se obtiene la evaluación de la función objetivo del mejor

individuo encontrado, su dimensión y el error de clasificación. En las siguientes tablas se resumen dichos resultados.

Tabla 3.3 Evaluación de la función objetivo para los once mapas.

Fármacos	AS	ACS	MMAS
apv	0.363557	0.264536	**0.235142**
atv	0.197633	**0.131279**	0.143482
idv	**0.150973**	0.242312	0.155946
lpv	0.105066	0.119267	**0.103126**
nfv	0.096988	0.094816	**0.084132**
rtv	0.16098	0.232676	**0.140587**
sqv	0.234437	0.193374	**0.15663**
ddi	0.384659	0.370677	**0.200503**
dlv	**0.195561**	0.222886	0.215791
efv	**0.088717**	0.153716	0.119144
nvp	0.107121	0.102283	**0.100889**

Por apreciación el que mejor comportamiento exhibe es el MMAS en cuanto a la evaluación de la función objetivo. Precisamente el test de Friedman es una prueba de varianza no paramétrica que permite estudiar la existencia o no de diferencias significativas entre los diferentes valores que conforman la muestra.

Los resultados (Figura 3.2) basados en la significación del test (menor que 0.05) rechazan la hipótesis principal, por lo que se puede concluir la existencia de diferencias significativas entre los grupos. Además, esta prueba ofrece los rangos medios (*Mean Rank*) coincidiendo el menor valor con MMAS, seguido por AS, y finalmente ACS.

Ranks

	Mean Rank
AS	2.27
ACS	2.36
MMAS	1.36

Test Statistics[a]

N			11
Chi-Square			6.727
df			2
Asymp. Sig.			.035
Monte Carlo Sig.	Sig.		.036
	99% Confidence Interval	Lower Bound	.031
		Upper Bound	.041

a. Friedman Test

Fig. 3.2 Test de Friedman para la función objetivo.

La existencia de diferencias significativas de forma general no es suficiente para determinar cuál optimizador es el mejor en cuanto a la evaluación de la función objetivo, sin embargo brinda una idea

del comportamiento de dichos algoritmos gracias a los rangos medios. Por tanto, es necesario que existan diferencias entre los algoritmos en comparaciones dos a dos.

Para ello se utiliza el test de Wilcoxon para muestras pareadas (Figura 3.3) con el cual es posible analizar este comportamiento. En este caso, la significación calculada para el par MMAS-ACS es menor que 0.05 por lo que se rechaza la hipótesis principal fundamental conservativa y se confirma la existencia de diferencias significativas a favor de MMAS y entre los pares MMAS-AS y AS-ACS la significación calculada es superior a 0.05 por lo que no existen diferencias significativas en estos pares. Por lo cual se puede concluir que:

- MMAS y AS muestran un comportamiento similar
- MMAS obtuvo mejor desempeño que ACS
- AS y ACS se comportan de forma similar

Test Statistics[a,c]

			MMAS - AS	MMAS - ACS	AS - ACS
Z			-1.689[b]	-2.578[b]	-.178[b]
Asymp. Sig. (2-tailed)			.091	.010	.859
Monte Carlo Sig. (2-tailed)	Sig.		.101	.005	.898
	99% Confidence Interval	Lower Bound	.093	.003	.890
		Upper Bound	.109	.007	.905
Monte Carlo Sig. (1-tailed)	Sig.		.051	.003	.447
	99% Confidence Interval	Lower Bound	.046	.001	.434
		Upper Bound	.057	.004	.460

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

c. Based on 10000 sampled tables with starting seed 624387341.

Fig. 3.3 Test de Wilcoxon para muestras pareadas en cuanto a función objetivo.

Recordemos que el objetivo principal del algoritmo propuesto es la determinación del conjunto mínimo de conceptos que mejor describa al sistema, y que en la ponderación de la función objetivo se tomó como 0.95 de importancia a la preservación del error y 0.05 de importancia a la reducción de la dimensión. Lo que trae consigo que a la hora de determinar el optimizador que ha tenido mejor desempeño, la validación utilizando la información brindada por la función objetivo provee una idea del comportamiento pero no es determinante para decidir. Por tanto, a continuación se muestra la información correspondiente a la reducción de la dimensión y preservación del error.

Tabla 3.4 Dimensión obtenida para los once mapas

Fármacos	AS	ACS	MMAS	Original
apv	24.8	**23.1**	23.8	51

atv	28.1	20.9	**15**	59
idv	22.5	**16.2**	17.2	48
lpv	21.5	**18.5**	22	51
nfv	18.1	13.8	**11.8**	40
rtv	19.3	**14.1**	14.2	44
sqv	21.8	19.8	**17**	48
ddi	15.9	**14.1**	15.2	38
dlv	21.1	15.8	**14.5**	44
efv	17.5	**14.1**	**14.1**	41
nvp	**10.5**	13.1	14.3	45

Tabla 3.5 Por ciento de reducción de la dimensión obtenida para los once mapas

Fármacos	AS	ACS	MMAS
apv	51.37255	**54.70588**	53.33333
atv	52.37288	64.57627	**74.57627**
idv	53.125	**66.25**	64.16667
lpv	57.84314	**63.72549**	56.86275
nfv	54.75	65.5	**70.5**
rtv	56.13636	**67.95455**	67.72727
sqv	54.58333	58.75	**64.58333**
ddi	58.15789	**62.89474**	60
dlv	52.04545	64.09091	**67.04545**
efv	57.31707	**65.60976**	**65.60976**
nvp	**76.66667**	70.88889	68.22222
Promedio	56.76094	64.08604	64.78428

A continuación se presentan los datos de reducción de la dimensión en gráficos más visuales para cada una de los once mapas. En la primera columna se muestra la dimensión original de los mapas y en las tres columnas restantes se muestra la dimensión obtenida por los tres optimizadores seleccionados.

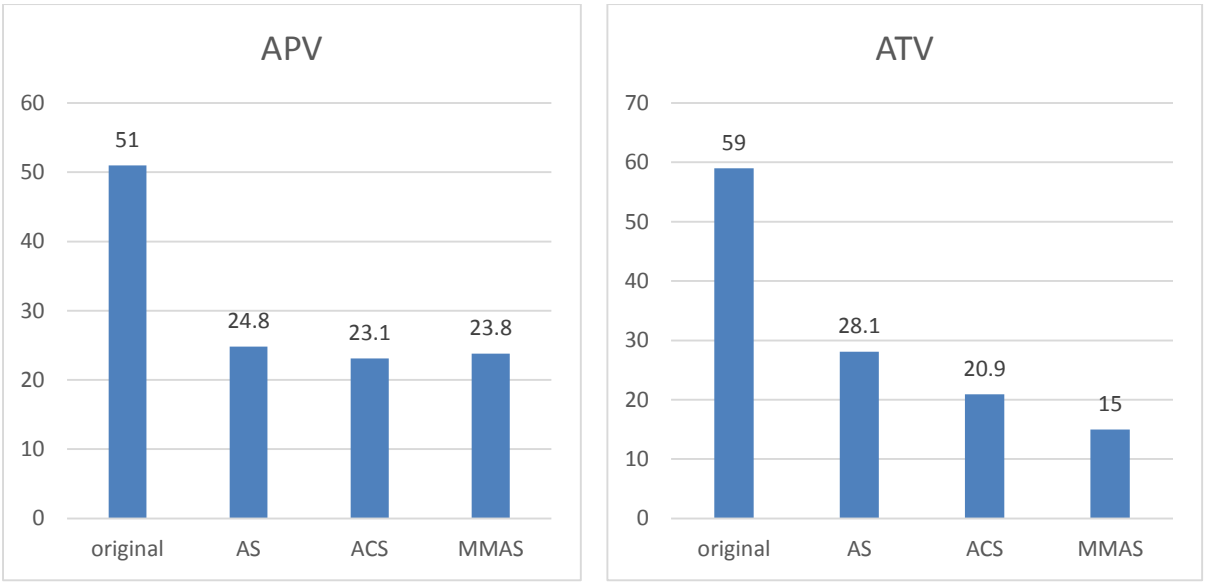


Fig. 3.4 Reducción de la dimensión obtenida para APV y ATV.

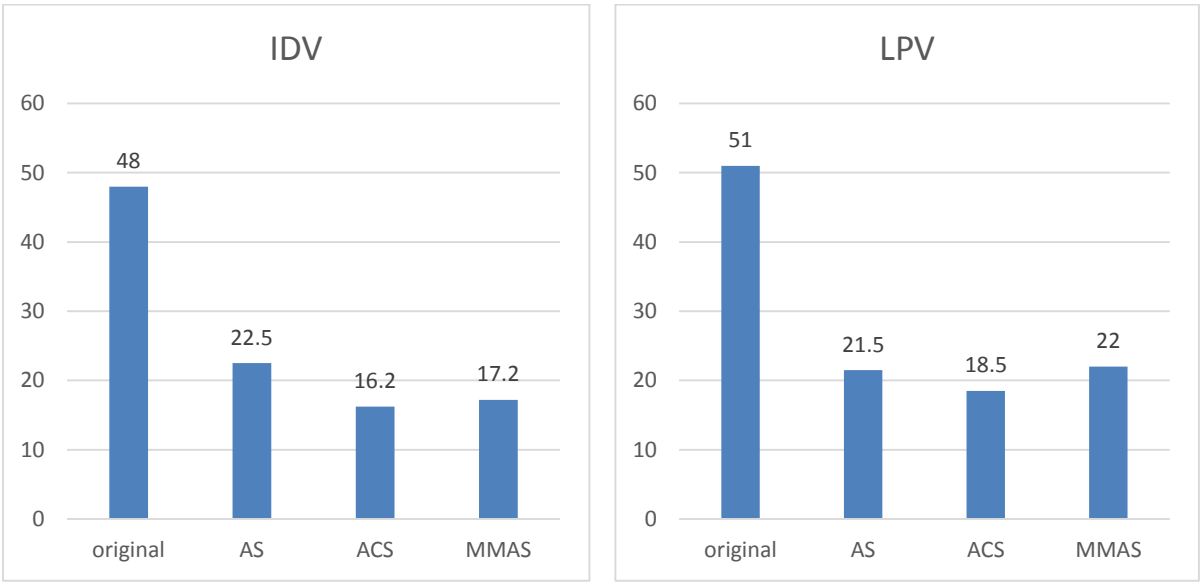


Fig. 3.5 Reducción de la dimensión obtenida para IDV y LPV.

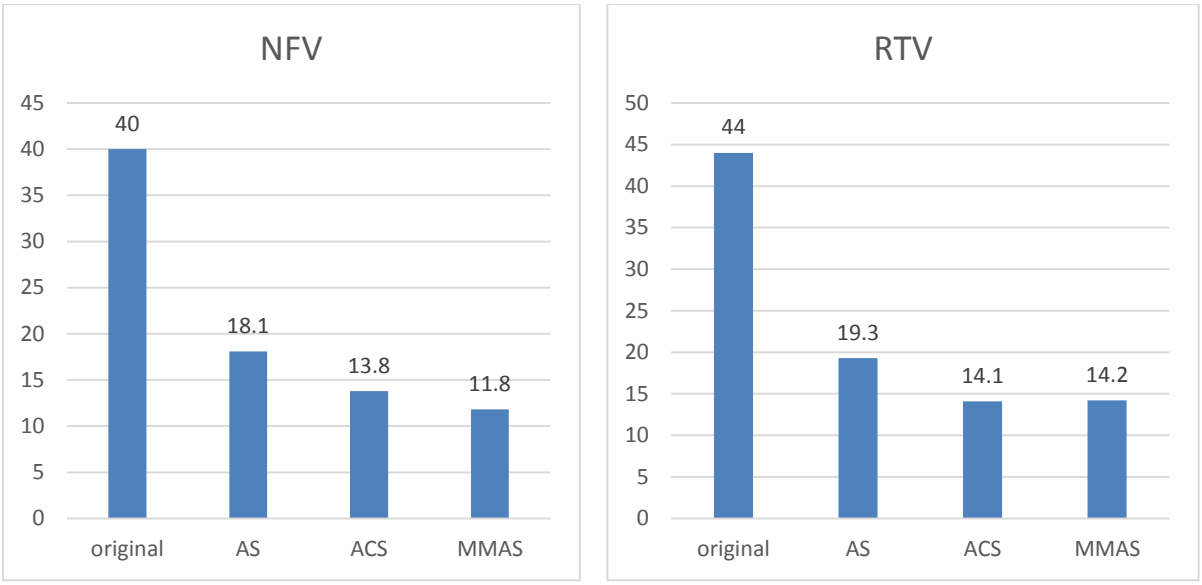


Fig. 3.6 Reducción de la dimensión obtenida para NFV y RTV.

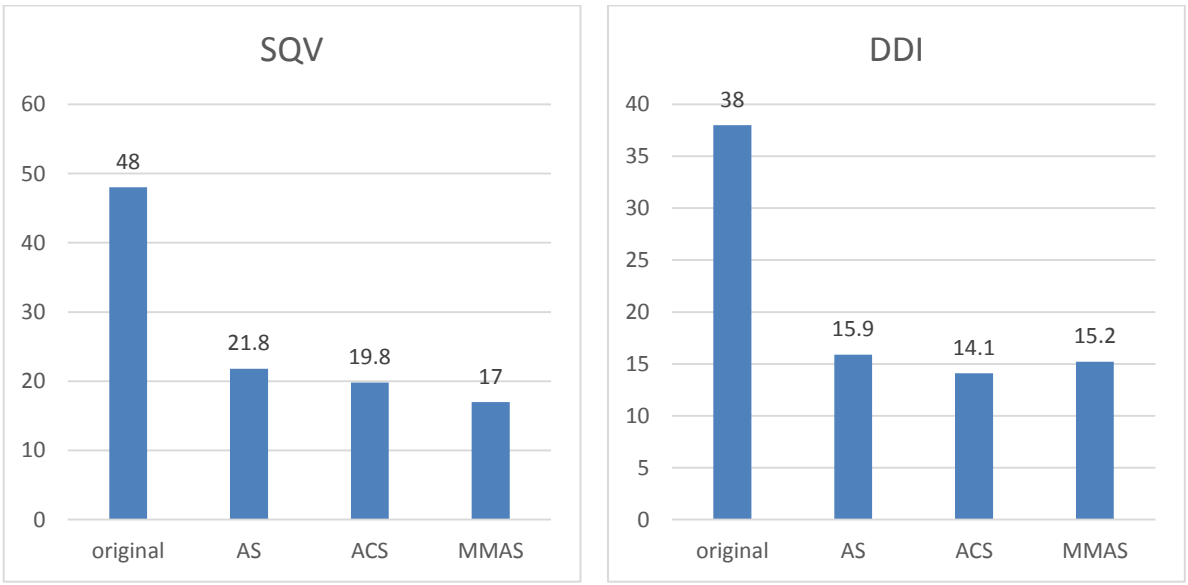


Fig. 3.7 Reducción de la dimensión obtenida para SQV y DDI.

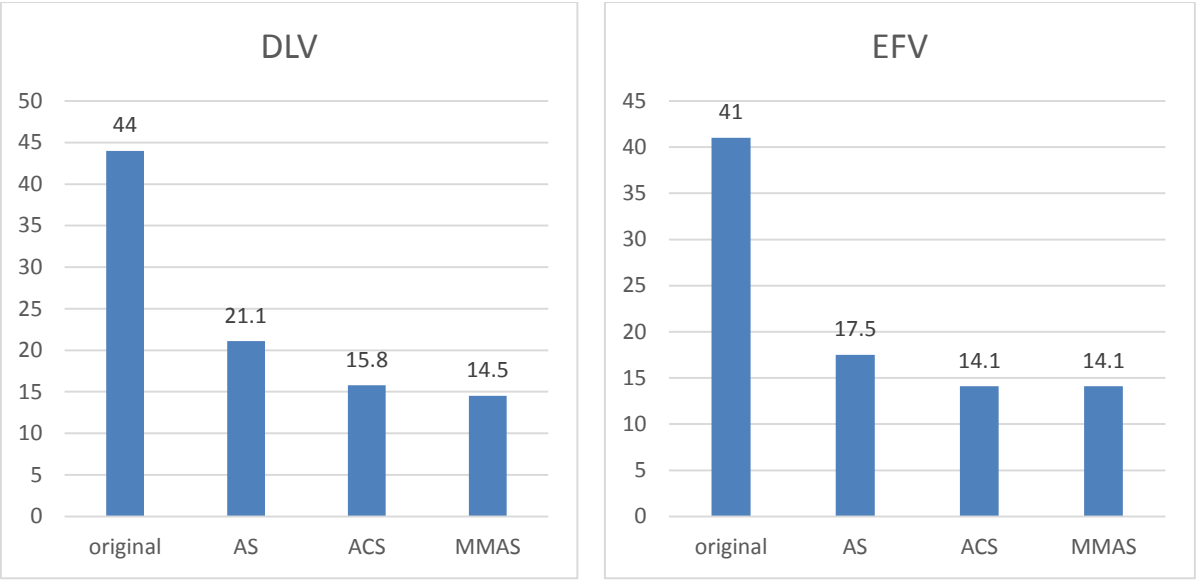


Fig. 3.8 Reducción de la dimensión obtenida para DLV y EFV.

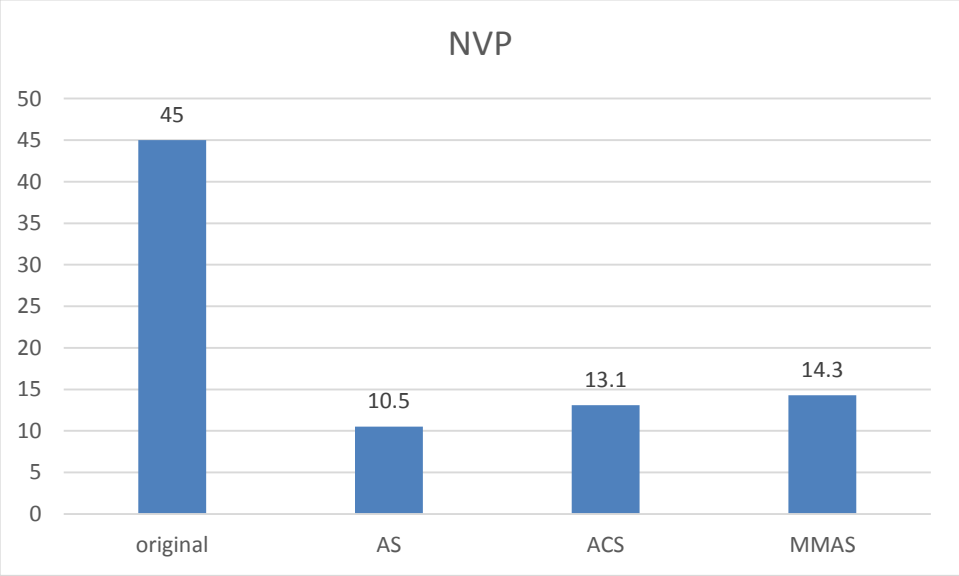


Fig. 3.9 Reducción de la dimensión obtenida para NVP.

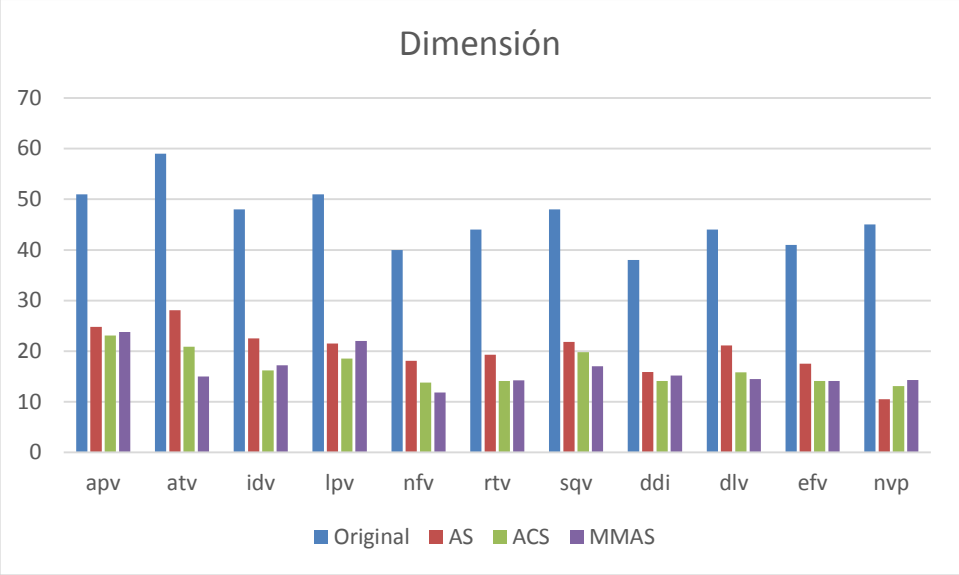


Fig. 3.10 Comportamiento general de la dimensión obtenida para los once mapas.

Para determinar la existencia o no de diferencias significativas entre los diferentes valores que conforman la muestra se aplica el test de Friedman (vea siguiente figura) en el cual el orden según los rangos medios de menor a mayor es ACS, MMAS, AS y finalmente la dimensión del mapa original.

Ranks

	Mean Rank
AS	2.73
ACS	1.50
MMAS	1.77
Original	4.00

Test Statistics[a]

N			11
Chi-Square			25.514
df			3
Asymp. Sig.			.000
Monte Carlo Sig.	Sig.		.000
	99% Confidence Interval	Lower Bound	.000
		Upper Bound	.000

a. Friedman Test

Fig. 3.11 Test de Friedman para la a reducción de la dimensión.

La existencia de diferencias significativas de manera grupal que indica Friedman no es suficiente para determinar cuál optimizador es el mejor en cuanto a la reducción de la dimensión. Por tanto se aplica el test de Wilcoxon (Figura 3.12) para muestras pareadas.

Test Statistics[a,c]

			ACS - MMAS	ACS - AS	MMAS - AS	AS - Original
Z			-.357[b]	-2.578[d]	-2.401[d]	-2.936[d]
Asymp. Sig. (2-tailed)			.721	.010	.016	.003
Monte Carlo Sig. (2-tailed)	Sig.		.768	.007	.013	.001
	99% Confidence Interval	Lower Bound	.757	.005	.010	.000
		Upper Bound	.779	.009	.016	.001
Monte Carlo Sig. (1-tailed)	Sig.		.390	.004	.007	.000
	99% Confidence Interval	Lower Bound	.377	.002	.004	.000
		Upper Bound	.402	.005	.009	.001

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

c. Based on 10000 sampled tables with starting seed 334431365.

d. Based on positive ranks.

Fig. 3.12 Test de Wilcoxon para muestras pareadas en cuanto a reducción de la dimensión.

Como se puede observar en la figura anterior:

La significación calculada para el par ACS-MMAS es superior a 0.05 por lo cual se puede determinar que no existen diferencias significativas en dicho par.

La significación calculada para los pares ACS-AS, MMAS-AS y AS-Original es menor que 0.05 por lo que se rechaza la hipótesis principal fundamental conservativa y se confirma la existencia de diferencias significativas entre estos pares a favor de ACS, MMAS y AS respectivamente; por lo cual se puede concluir en cuanto al desempeño de los optimizadores en la reducción de la dimensión que:

- MMAS y ACS mostraron un desempeño similar.
- MMAS y ACS mostraron mejor desempeño que AS.
- El AS tuvo diferencias significativas respecto a la dimensión original, a favor de AS y por tanto MMAS y ACS también poseen dicho comportamiento.

El último aspecto por tratar está relacionado con la preservación del error. A continuación se efectúa la validación estadística de dichos resultados.

Tabla 3.3 Error obtenido para los once fármacos.

	AS	ACS	MMAS	Original
apv	0.24359	0.210256	**0.205128**	0.20513
atv	0.097122	0.092806	**0.092086**	0.08633
idv	0.109326	0.119171	**0.104663**	0.10363
lpv	0.077368	0.079474	**0.074737**	0.07895
nfv	0.078277	0.081648	**0.073034**	0.07491
rtv	0.070046	0.075115	**0.067742**	0.05991
sqv	0.150251	0.140704	**0.126131**	0.13568
ddi	0.228205	0.220513	**0.169231**	0.17949
dlv	**0.093631**	0.095541	0.096815	0.0828

efv	**0.070922**	0.073759	0.07234	0.06383
nvp	0.100478	0.092344	**0.089474**	0.09091

Tabla 3.4 Diferencia del error obtenido con el error original.

Fármacos	AS	ACS	MMAS
apv	0.03846	0.005126	**-0.000002**
atv	0.010792	0.006476	**0.005756331**
idv	0.005696	0.015541	**0.001033212**
lpv	-0.00158	0.000524	**-0.00421315**
nfv	0.003367	0.006738	**-0.00187629**
rtv	0.010136	0.015205	**0.007831935**
sqv	0.014571	0.005024	**-0.00954934**
ddi	0.048715	0.041023	**-0.01025923**
dlv	**0.010831**	0.012741	0.014015287
efv	**0.007092**	0.009929	0.008510426
nvp	0.009568	0.001434	**-0.00143631**

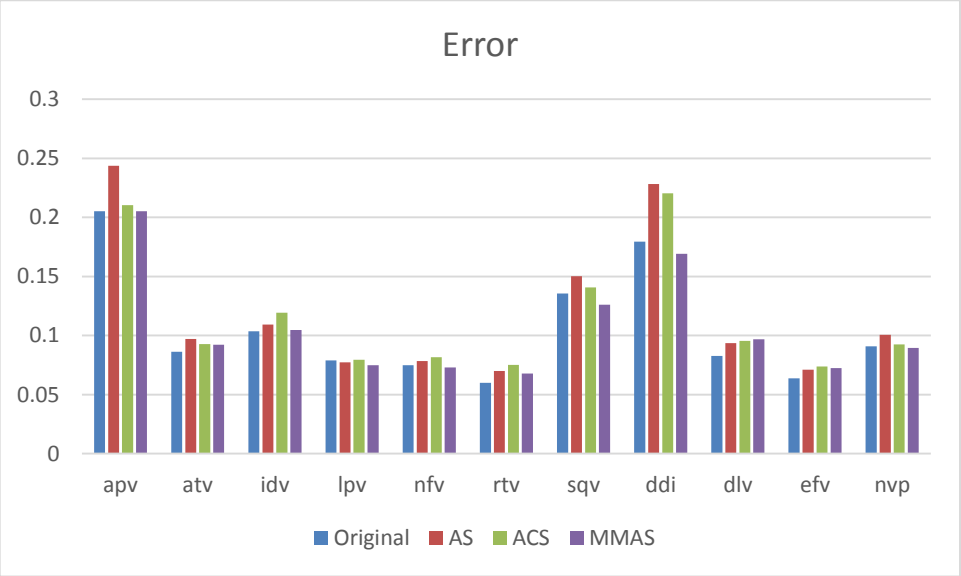


Fig. 3.13 Comportamiento general del error en los once mapas.

Como se puede observar en las tablas mostradas anteriormente el optimizador que mejor desempeño muestra es el MMAS. Para determinar la existencia o no de diferencias significativas entre los diferentes valores que conforman la muestra se aplica el test de Friedman (vea siguiente figura) en el que el orden según los rangos medios es Original, MMAS, AS y finalmente ACS.

Ranks

	Mean Rank
AS	3.18
ACS	3.45
MMAS	1.73
Original	1.64

Test Statistics[a]

N			11
Chi-Square			17.945
df			3
Asymp. Sig.			.000
Monte Carlo Sig.	Sig.		.000
	99% Confidence Interval	Lower Bound	.000
		Upper Bound	.000

a. Friedman Test

Fig. 3.14 Test de Friedman en cuanto la preservación del error.

La existencia de diferencias significativas de manera grupal que indica Friedman no es suficiente para determinar cuál optimizador es el mejor en cuanto a la preservación del error. Por tanto se aplica el test de Wilcoxon (Figura 3.15) para muestras pareadas.

Test Statistics[a,c]

			AS - Original	ACS - Original	MMAS - Original
Z			-2.845[b]	-2.934[b]	-.089[b]
Asymp. Sig. (2-tailed)			.004	.003	.929
Monte Carlo Sig. (2-tailed)	Sig.		.002	.000	.966
	99% Confidence Interval	Lower Bound	.001	.000	.961
		Upper Bound	.003	.001	.970
Monte Carlo Sig. (1-tailed)	Sig.		.001	.000	.484
	99% Confidence Interval	Lower Bound	.000	.000	.471
		Upper Bound	.002	.001	.497

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

c. Based on 10000 sampled tables with starting seed 624387341.

Fig. 3.15 Test de Wilcoxon para muestras pareadas en cuanto la preservación del error.

Como se puede apreciar en la figura anterior los pares AS-Original y ACS-Original tienen un nivel de significación inferior a 0.05 lo cual asegura que las diferencias en estos pares son relevantes y a favor de Original, lo cual conlleva a que los optimizadores AS y ACS no preservan el error. El par MMAS-Original tiene un nivel de significación superior a 0.05 lo que implica que no existen diferencias relevantes entre el error obtenido por MMAS y el error original; por lo tanto el optimizador MMAS preserva el error.

3.4 Conclusiones parciales

La validación del algoritmo propuesto se realizó a través de un problema de clasificación real referente a dos proteínas del VIH. Se modelo las proteínas del VIH utilizando MCD y el ajuste de las causalidades se logró mediante la aplicación de un esquema de aprendizaje colectivo.

Se puede determinar gracias a las pruebas comparativas (tabla 3.5) realizadas, que el algoritmo propuesto utilizando como optimizador a MMAS permite obtener un conjunto mínimo de conceptos que describen el sistema modelado y que preservan el error. Además de obtener un porciento de reducción promedio de la dimensión de los mapas del 65%.

Tabla 3.5 Mejor desempeño de los optimizadores validado estadísticamente.

Optimizador	Evaluación de la función objetivo	Reducción de la dimensión	Preservación del Error
AS			
ACS		X	
MMAS	X	X	X

Por lo tanto como última conclusión se confirma la hipótesis de investigación de esta propuesta.

CONCLUSIONES

Los algoritmos de aprendizaje para MCD están enfocados a ajustar la causalidad del mapa y el algoritmo tratado en esta propuesta hasta donde es conocimiento del autor no ha sido abordado en la literatura existente.

El algoritmo de aprendizaje propuesto permite optimizar los MCD a través de la determinación de un subconjunto mínimo de conceptos que describen al mapa garantizando que el mapa resultante de la optimización preserva su calidad. Para ello se definieron cinco aspectos esenciales del mismo.

Se validaron estadísticamente los resultados obtenidos por el algoritmo de aprendizaje utilizando tres optimizadores confirmando que la determinación de un optimizador potente es esencial en el desempeño del método de búsqueda del algoritmo propuesto.

Se optimizó la topología de MCD a través de la reducción de los conceptos no relevantes. Lo cual permitió una fácil interpretación del sistema modelado.

RECOMENDACIONES

1. Estudiar el impacto de otros optimizadores en el algoritmo de aprendizaje propuesto.
2. Estudiar el desempeño del algoritmo propuesto con otros casos de estudios, por ejemplo: el problema del transporte.

Referencias Bibliográficas

AGUILAR, J. 2005. A survey about fuzzy cognitive maps papers. *International journal of computational cognition,* 3**,** 27-33.

ALONSO, S., CORDÓN, O., FERNÁNDEZ, I. & HERRERA, F. 2001. La metaheurística de optimización basada en colonias de hormigas: modelos y nuevos enfoques. *Optimización inteligente: técnicas de inteligencia computacional para optimización***,** 261-314.

AXELROD, R. M., STUDIES, I. O. P. P. & STUDIES, I. O. I. 1976. *Structure of decision: The cognitive maps of political elites*, Princeton university press Princeton.

BEERENWINKEL, N., SCHMIDT, B., WALTER, H., KAISER, R., LENGAUER, T., HOFFMANN, D., KORN, K., & SELBIG, J 2002. Diversity and complexity of HIV-1 drug resistance: a bioinformatics approach to predicting phenotype from genotype. *Proceedings of the National Academy of Sciences of the United States of America,* 99**,** 8271-6.

BELLO, R. & PURIS, A. 2008. Feature Selection through Dynamic Mesh Optimization.

CARVALHO, J. P. & TOMÉ, J. Fuzzy mechanisms for causal relations. Proceedings of the Eighth International Fuzzy Systems Association World Congress, IFSA, 1999. 1009-1013.

DENEUBOURG, J.-L., ARON, S., GOSS, S. & PASTEELS, J. M. 1990. The self-organizing exploratory pattern of the argentine ant. *Journal of insect behavior,* 3**,** 159-168.

DICKERSON, J. A. & KOSKO, B. Virtual worlds as fuzzy cognitive maps

Virtual Reality Annual International Symposium, 1993., 1993 IEEE, 1993. IEEE, 471-477.

DORIGO, M. & GAMBARDELLA, L. M. 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on,* 1**,** 53-66.

DORIGO, M., MANIEZZO, V. & COLORNI, A. 1996. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics -Part B,* 26: 1**,** 29-41.

DORIGO, M. & STÜTZLE, T. 2011. The ant colony optimization metaheuristic: Algorithms, applications, and advances. *Handbook of metaheuristics.* Springer.

EDELKAMP, S. & SCHROEDL, S. 2011. *Heuristic search: theory and applications*, Elsevier.

HUERGA, A. V. A balanced differential learning algorithm in fuzzy cognitive maps. Proceedings of the 16th International Workshop on Qualitative Reasoning, 2002.

HUFF, A. 1990. Mapping Strategic Thought.

KHAN, M. S. & CHONG, A. Fuzzy Cognitive Map Analysis with Genetic Algorithm. IICAI, 2003. 1196-1205.

KHAN, M. S. & QUADDUS, M. 2004. Group Decision Support using Fuzzy Cognitive Maps for Causal Reasoning. *Group Decision and Negotiation Journal,* 13**,** 463-480.

KIERCZAK, M., DRAMISKI, M., & KORONACKI, J. 2009. A Rough set-Based Model of HIV-1 Reverse Transcriptase Resistome. *Bioinformatics and Biology Insights,* 3**,** 109-127.

KOSKO, B. 1986. Fuzzy cognitive maps. *International journal of man-machine studies,* 24**,** 65-75.

KOSKO, B. 1988. Hidden patterns in combined and adaptive knowledge networks. *International Journal of Approximate Reasoning,* 2**,** 377-393.

KOSKO, B. 1992. *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence*, Prentice-Hall, Inc.

KOULOURIOTIS, D., DIAKOULAKIS, I. & EMIRIS, D. Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior. Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, 2001. IEEE, 364-371.

LEÓN, M., RODRIGUEZ, C., NÁPOLES, G., BELLO, R. & VANHOOF, K. 2010. Uso de Mapas Cognitivos Difusos para modelar Representaciones Mentales que caracterizan el Comportamiento de Viajes. *Segundo Taller Cubano Eureka 2010*

MCNEILL, F. M. & THRO, E. 1994. *Fuzzy logic: a practical approach*, Academic Press Professional, Inc.

MIYAZAWA, S., JERNIGAN, R.L 1999. Contacts energies Self-Consistent Estimation of Inter-Residue Protein Contact Energies Based on an Equilibrium Mixture Approximation of Residues. PROTEINS: Structure, Function, and Genetics. 34**,** 49-68.

NÁPOLES, G. & GRAU, I. 2013. Sequence positions related with drug resistance for the HIV-1 protease and reverse transcriptase proteins: a brief review from literature.

NÁPOLES, G., GRAU, I. & BELLO, R. 2012a. Constricted Particle Swarm Optimization based algorithm for global optimization. *POLIBITS Research journal on Computer science and computer engineering with applications,* 46**,** 5-11.

NÁPOLES, G., GRAU, I. & BELLO, R. 2012b. Particle Swarm Optimization with Random Sampling in Variable Neighbourhoods for solving Global Minimization Problems. *In: Christensen, A.L. (eds.) ANTS 2012. LNCS,* 7461.

NÁPOLES, G., GRAU, I., BELLO, R. & GRAU, R. 2014. Two-steps learning of Fuzzy Cognitive Maps for prediction and knowledge discovery on the HIV-1 drug resistance. *Expert Systems with Applications*.

NÁPOLES, G., GRAU, I., PÉREZ-GARCÍA, R. & BELLO, R. Learning of Fuzzy Cognitive Maps for simulation and knowledge discovery. Fourth International Workshop on Knowledge Discovery, Knowledge Management and Decision Support, 2013. Atlantis Press.

PAPAGEORGIOU, E., STYLIOS, C. & GROUMPOS, P. 2003. Fuzzy cognitive map learning based on nonlinear Hebbian rule. *AI 2003: Advances in Artificial Intelligence.* Springer.

PAPAGEORGIOU, E., STYLIOS, C. D. & GROUMPOS, P. P. 2004. Active Hebbian learning algorithm to train fuzzy cognitive maps. *International Journal of Approximate Reasoning,* 37**,** 219-249.

PAPAGEORGIOU, E. I. 2011. Review study on Fuzzy Cognitive Maps and their applications during the last decade. *IEEE International Conference on Fuzzy Systems.*

PAPAGEORGIOU, E. I. 2012. Learning algorithms for fuzzy cognitive maps—A review study. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* 42**,** 150-163.

PAPAGEORGIOU, E. I. & GROUMPOS, P. P. 2005. A new hybrid method using evolutionary algorithms to train fuzzy cognitive maps. *Applied Soft Computing,* 5**,** 409-431.

PARSOPOULOS, K. E., PAPAGEORGIOU, E. I., GROUMPOS, P. & VRAHATIS, M. N. A first study of fuzzy cognitive maps learning using particle swarm optimization. Evolutionary Computation, 2003. CEC'03. The 2003 Congress on, 2003. IEEE, 1440-1447.

STACH, W., KURGAN, L. & PEDRYCZ, W. 2005a. A survey of fuzzy cognitive map learning methods. *Issues in soft computing: theory and applications***,** 71-84.

STACH, W., KURGAN, L. & PEDRYCZ, W. Data driven nonlinear Hebbian learning method for fuzzy cognitive maps. IEEE World Congress Computation Intelligence., 2008 Hong Kong. IEEE.

STACH, W., KURGAN, L. & PEDRYCZ, W. 2010. Expert-based and computational methods for developing fuzzy cognitive maps. *Fuzzy Cognitive Maps.* Springer.

STACH, W., KURGAN, L., PEDRYCZ, W. & REFORMAT, M. 2005b. Genetic learning of fuzzy cognitive maps. *Fuzzy sets and systems,* 153**,** 371-401.

STACH, W. J. 2010. *Learning and Aggregation of Fuzzy Cognitive Maps: An Evolutionary Approach.* University of Alberta.

STÜTZLE, T. & HOOS, H. 1996. Improving the Ant System: A detailed report on the MAX-MIN Ant System.

STÜTZLE, T. & HOOS, H. H. 2000. MAX–MIN ant system. *Future generation computer systems,* 16**,** 889-914.

STÜTZLE, T. G. 1999. *Local search algorithms for combinatorial problems: analysis, improvements, and new applications*, Infix Sankt Augustin, Germany.

TSADIRAS, A. K. 2008. Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. *Information Sciences,* 178**,** 3880-3894.

WANG, L., PICHLER, E. E. & ROSS, J. 1990. Oscillations and chaos in neural networks: an exactly solvable model. *Proceedings of the National Academy of Sciences,* 87**,** 9467-9471.

ZHU, Y. & ZHANG, W. An integrated framework for learning fuzzy cognitive map using RCGA and NHL algorithm. Proceedings of 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM, 2008.