

Universidad Central "Marta Abreu" de las Villas.
Facultad Matemática Física y Computación
Licenciatura en Ciencia de la Computación



TRABAJO DE DIPLOMA

*Algoritmo de aprendizaje para mejorar la convergencia de los
Mapas Cognitivos Difusos sin afectar la precisión*

Autor: Leonardo Javier Concepción Pérez

Tutores: MSc. Gonzalo Nápoles, MSc. Isel Grau

Santa Clara

2016



Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

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

Firma del tutor

Firma del jefe del Laboratorio

“Si amas lo que haces, nunca será un trabajo”

Confucio

A quienes me lo han dado todo.

Agradezco sinceramente:

- A mi tutor MSc. Gonzalo Nápoles, por ser guía, ejemplo de profesional, rival en el deporte y amigo.
- A mi tutora MSc, Isel Grau, que salió del banquillo cuando llega la parte que no le gusta a ningún cibernético (escribir la tesis).
- A mis padres por su apoyo incondicional, su ánimo, sus consejos, su atención y su cariño.
- A toda mi familia que está siempre al tanto.
- A mi primo (El Bracamán) por su insistencia y sus preguntas sin respuesta.
- A mis amigos que siempre ayudan a despejar, a olvidar los problemas y a reírme de ellos, para luego, a veces, aprender un poquito.
- A todo aquel que ha influido en mi formación: mis maestras de primaria, mis profesoras y profesores de secundaria e IPVCE.
- A los compañeros que siguen y a los que no están, a los que su nombre ya no es el mismo, a quienes se acuerdan de mí a cada rato y a los que siempre están a mi lado.

RESUMEN

Los Mapas Cognitivos Difusos fueron diseñados por Kosko como una extensión de los Mapas Cognitivos. Existen varios algoritmos de aprendizaje supervisados y no supervisados para ajustar los parámetros que definen el comportamiento de estas estructuras basadas en el conocimiento. Sin embargo, los mapas ajustados que utilizan funciones continuas (o construidos por expertos) regularmente no son capaces de converger a un estado de equilibrio, ya que los estados inestables son frecuentes en sistemas continuos; limitando su eficiencia en problemas de simulación de escenarios. Recientemente se han publicado métodos analíticos para determinar condiciones de convergencia en Mapas Cognitivos Difusos Sigmoideos, pero los resultados no son aplicables en contextos de simulación de múltiples escenarios.

Este trabajo propone un algoritmo de aprendizaje heurístico para mejorar las propiedades de convergencia en Mapas Cognitivos Difusos Sigmoideos preservando la precisión del sistema, una vez que se han estimado los pesos causales. El algoritmo se reduce a un problema de optimización donde se calculan la inclinación y el desplazamiento de cada función sigmoidea (una por cada concepto del mapa), minimizando la variabilidad en las respuestas para todas las instancias y penalizando la inestabilidad y la convergencia a puntos fijos únicos del sistema. Para solucionar este problema se adopta la Optimización de Enjambres de Partículas, y se evalúa el desempeño del algoritmo usando un conjunto de datos artificiales generado específicamente para ello. También se establecen algunas definiciones y propiedades relacionadas con la estabilidad y precisión en los Mapas Cognitivos Difusos Sigmoideos, permitiendo otro enfoque de evaluación para el algoritmo propuesto.

ABSTRACT

Fuzzy Cognitive Maps were designed by Kosko as an extension of Cognitive Maps. There are several supervised and non-supervised learning algorithms to adjust the parameters that define the behavior of these structures based on knowledge. However, adjusted maps using continuous functions (or designed by experts) are regularly unable to converge to an equilibrium state, given that unstable states are frequent in continuous systems. As a result, Fuzzy Cognitive Maps efficiency in scenario simulation problems is limited. Analytical methods for determining convergence conditions in Sigmoid Fuzzy Cognitive Maps has been published, but the results are not applicable in the context of multiple scenarios simulation.

This paper proposes a heuristic learning algorithm to improve the convergence properties in Sigmoid Fuzzy Cognitive Maps preserving the system accuracy, once estimation of causal weights is done. The algorithm leads to an optimization problem where the inclination and the displacement of each sigmoid function (one for each map concept) are calculated by minimizing the variability in responses for all instances and penalizing instability and convergence to unique equilibrium states. To solve this problem, Particle Swarms Optimization is adopted, and the performance of the algorithm is evaluated using a set of artificial data generated specifically for this task. Some definitions and properties, related to stability and accuracy in Sigmoid Fuzzy Cognitive Maps are also introduced, allowing another assessment approach for the proposed algorithm.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
Objetivo general	3
Objetivos específicos	3
Tareas de investigación	3
Hipótesis de investigación	4
CAPÍTULO 1. TEORÍA DE MAPAS COGNITIVOS DIFUSOS	6
1.1 Mapas Cognitivos Difusos.....	6
1.2 Proceso de inferencia de los Mapas Cognitivos Difusos	7
1.3 Algoritmos de aprendizaje para Mapas Cognitivos Difusos	10
1.3.1 Algoritmos de aprendizaje Hebbianos	10
1.3.2 Algoritmos de aprendizaje basados en metaheurísticas.....	13
1.4 Mapas cognitivos difusos para modelación y clasificación	16
1.4.1 Modelación.....	17
1.4.2 Clasificación	20
1.5 Estabilidad en mapas cognitivos difusos	22
1.6 Consideraciones finales del capítulo	27
CAPÍTULO 2. ALGORITMO PARA MEJORAR LA CONVERGENCIA EN MAPAS COGNITIVOS DIFUSOS SIGMOIDEOS.....	29
2.1 Preliminares sobre el algoritmo	29
2.2 Descripción de la propuesta	30
2.3 Selección del optimizador	34
2.3.1 Optimización de Enjambres de Partículas	34
2.3.2 PSO con topologías estructuradas.....	37
2.4 Definiciones sobre convergencia y precisión	38
2.4.1 Mapas Cognitivos Difusos ε -estables y ε -inestables	39
2.5 Conclusiones parciales.....	47
CAPÍTULO 3. EVALUACIÓN DEL ALGORITMO DE APRENDIZAJE	50
3.1 Generación del conjunto de mapas de prueba	50
3.2 Validación experimental.....	51
3.2.1 Selección de topología del PSO	53
3.2.2 Selección del parámetro α para la función objetivo	56
3.3 Validación con resultados matemáticos.....	59
3.3.1 Analizando E -estabilidad.....	60

3.3.2	Analizando E -inestabilidad.....	63
3.4	Conclusiones parciales.....	66
CONCLUSIONES		67
RECOMENDACIONES		68
REFERENCIAS BIBLIOGRÁFICAS		69

INTRODUCCIÓN

Los Mapas Cognitivos Difusos (MCD) fueron introducidos por (Kosko 1984) basado en el trabajo de (Axelrod 1976) sobre los mapas cognitivos. Los MCD son redes de inferencia que usan grafos dirigidos cíclicos que representan las relaciones causales entre los conceptos. Para ilustrar los diferentes aspectos relacionados con el comportamiento del sistema, un MCD consiste en un conjunto de nodos donde cada uno representa una característica del sistema. Las interacciones entre los nodos representan la dinámica del sistema. Debido a su flexibilidad para representar sistemas dinámicos, podemos encontrar ejemplos de su aplicación en ciencia política, en el campo económico, representando conocimiento científico social y describiendo métodos de toma de decisiones. (Boutalis et al. 2009).

Existen dos estrategias fundamentales para diseñar un MCD: manual o automática. En la primera variante, expertos humanos que conocen el comportamiento del sistema, determinan los conceptos que lo describen, así como la dirección e intensidad de las conexiones causales entre las neuronas. En la segunda variante la topología del mapa y la matriz de pesos se construyen a partir de datos históricos, o ejemplos representativos que definen el comportamiento del sistema. Independientemente de la estrategia, los expertos del dominio necesitan definir cuál es la arquitectura que mejor se ajusta al problema que se pretende modelar, así como las restricciones inherentes al modelo. Hasta donde es conocido, no existe una topología única capaz de modelar óptimamente cualquier problema (Nápoles 2014).

Los MCD simulan su evolución a través del tiempo, dando respuestas directas a preguntas causales del tipo “*what-if*” por lo que la robustez y precisión de las respuestas dependen en gran medida de la matriz de pesos. El proceso de estimación de los valores exactos de los pesos causales de un MCD es una tarea complicada para un ser humano, reduciendo la usabilidad de estas estructuras en problemas reales. Por esto los algoritmos de aprendizaje que modifican la matriz de pesos han demostrado ser componentes esenciales para mejorar la operación y precisión de los MCD. El análisis y desarrollo de estos algoritmos se concentra principalmente en los pesos, que son actualizados o modificados basados en tres fuentes de conocimiento: datos históricos, conocimiento de expertos del dominio o una combinación del conocimiento de los expertos y los datos históricos. Proponer algoritmos de aprendizaje eficientes para MCD constituye un campo

abierto de investigación (Papageorgiou 2012). Por ejemplo, las funciones de evaluación utilizadas en el aprendizaje son frecuentemente no convexas y caracterizadas por espacios de búsqueda multimodales, donde los métodos de aprendizajes son propensos a converger prematuramente hacia soluciones locales.(Nápoles et al. 2013) En síntesis, el diseño de un MCD requiere procedimientos más escalables y robustos, que sean capaces de detectar posibles soluciones locales y donde no se requiera la intervención del experto.

Una vez que se han estimado los valores causales entre los conceptos (manualmente o de forma automatizada), el experto puede simular la dinámica del sistema investigado. En la primera etapa las neuronas son activadas, y luego el proceso de inferencia actualiza de forma iterativa el valor de activación de cada concepto. Este algoritmo finaliza cuando el mapa converge a un estado de equilibrio (el mapa produce la misma salida a través del tiempo) o se alcanza un número máximo de iteraciones. En el primer caso se dice que el mapa es estable, mientras que el segundo caso el mapa puede tener un comportamiento cíclico o caótico. Los estados inestables son frecuentes en sistemas continuos, limitando la eficiencia de estas estructuras neuronales en problemas de toma de decisiones o clasificación de patrones. (Nápoles, Bello, et al. 2014)

Hasta donde el autor conoce, no existe un procedimiento que permita producir matrices de pesos estables, sin afectar la exactitud del sistema resultante. Incluso en (Kosko 1998) se afirma que este problema podría permanecer muchos años sin una solución matemática precisa, aunque recientemente se han publicado algunos trabajos que establecen condiciones de convergencia en sistemas con funciones de activación continuas (ej. variantes sigmoideas).

Por ejemplo, en (Boutalis et al. 2009) los autores demostraron que si la matriz de pesos satisface ciertas premisas entonces el mapa siempre convergerá a un único punto fijo, independientemente del estado inicial de los conceptos. De forma similar, (Knight, Lloyd & Penn 2014) demostraron que si la inclinación de la función de activación sigmoidea es suficientemente pequeña, entonces el mapa tendrá una única solución linealmente estable. Sin embargo, estos resultados no son aplicables en contextos de clasificación de patrones, sino a MCD usados en tareas de simulación, y las razones serán argumentadas durante el desarrollo del presente trabajo.

OBJETIVO GENERAL

Desarrollar un algoritmo de aprendizaje supervisado, basado en las propiedades analíticas de las funciones de normalización, para mejorar la convergencia de los MCD Sigmoides usados en problemas de simulación de escenarios, preservando la precisión.

OBJETIVOS ESPECÍFICOS

1. Diseñar un algoritmo de aprendizaje para mejorar la convergencia de los MCD Sigmoides utilizando las propiedades analíticas de las funciones de normalización, que permita preservar la precisión del sistema.
2. Implementar el algoritmo de aprendizaje supervisado como un problema de optimización continuo.
3. Evaluar el desempeño del algoritmo utilizando un conjunto de datos artificiales con diferentes características.

TAREAS DE INVESTIGACIÓN

1. Realizar una revisión sobre los algoritmos de aprendizaje más representativos (supervisados y no supervisados) para Mapas Cognitivos Difusos.
2. Realizar una revisión sobre la aplicación de los Mapas Cognitivos Difusos en problemas de clasificación y modelación.
3. Identificar las principales deficiencias y potencialidades de los enfoques analíticos y metaheurísticos existentes para estabilizar Mapas Cognitivos Difusos continuos.
4. Identificar las propiedades ajustables de las funciones de normalización y modelar el nuevo algoritmo de aprendizaje como un problema de optimización de estas variables.
5. Seleccionar una metaheurística poblacional adecuada como optimizador.
6. Formalizar matemáticamente definiciones y propiedades relacionadas con la estabilidad y precisión en los MCD Sigmoides que permitan validar los resultados obtenidos por el algoritmo.
7. Generar un conjunto de prueba de mapas artificiales y sus escenarios con diferente grado de complejidad en la topología.

8. Evaluar el desempeño del algoritmo en problemas de simulación de escenarios, usando el conjunto de mapas generados.

HIPÓTESIS DE INVESTIGACIÓN

Es posible desarrollar un algoritmo de aprendizaje que utilice las propiedades analíticas de las funciones de normalización para mejorar la convergencia en los Mapas Cognitivos Difusos usados en problemas de simulación de escenarios, sin afectar la precisión.

CAPÍTULO 1

TEORÍA DE MAPAS
COGNITIVOS DIFUSOS

CAPÍTULO 1. TEORÍA DE MAPAS COGNITIVOS DIFUSOS

En este capítulo se aborda la teoría de los MCD desde su introducción en 1984 por Bart Kosko. Este capítulo además incluye la definición y explicación del proceso de inferencia, así como los posibles estados alcanzables respecto a la convergencia. Además, se hacen revisiones sobre las características de los MCD usados en problemas de clasificación y modelación, y sobre los principales métodos de aprendizaje reportados en la literatura, así como las ventajas y deficiencias de estos últimos. Finalmente se muestran los resultados analíticos más significativos reportados en la literatura sobre la convergencia de los MCD y se explican sus limitaciones.

1.1 MAPAS COGNITIVOS DIFUSOS

En (Axelrod 1976) se introducen los mapas cognitivos con el objetivo de representar el conocimiento científico social. Estos mapas son dígrafos con signo, donde los nodos representan conceptos variables (como la inestabilidad social, no como la sociedad) y las aristas son conexiones causales (más complicadas que las implicaciones lógicas). (Kosko 1986). Los conceptos representan variables, entidades o estados que usualmente describen el sistema que se pretende modelar en un dominio determinado. Por su parte las relaciones causales denotan conexiones entre los conceptos. Cada relación tiene un signo asociado para expresar la dirección de la causalidad. Más explícitamente, se cumple que:

- Si el signo de la conexión es positivo, entonces una variación en el concepto *causa* (concepto inicial) provocará una variación en el concepto *efecto* en la misma dirección.
- Si el signo de la relación es negativo, entonces una variación en el concepto *causa* provocará una variación en el concepto *efecto* en la dirección opuesta.
- Si la relación tiene valor 0 (o no existe), entonces no existe una relación causal entre los conceptos y una variación en el concepto *causa* no provocará variaciones en el concepto *efecto*.

Teniendo en cuenta solo las direcciones de las relaciones de causalidad es complicado modelar la semántica de un sistema complejo, por lo que los Mapas Cognitivos demostraron ser bastante limitados, y fue entonces necesaria la introducción por (Kosko 1984) de los Mapas Cognitivos Difusos. En los MCD a cada relación del mapa se le asigna un peso denotando el grado de

causalidad (estos pesos son valores numéricos en el intervalo $[-1, 1]$), que permite establecer la dirección e intensidad de la relación en cuestión.

De forma general la interpretación de las influencias causales entre los conceptos de un MCD es similar a la semántica de los Mapas Cognitivos. Para dos conceptos C_i y C_j se cumple que:

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

Más formalmente, desde el punto de vista matemático, un MCD puede ser definido usando una 4-tupla (C, W, A, f) donde $C = \{C_1, C_2, C_3, \dots, C_N\}$ representa el conjunto de conceptos del grafo, $W: (C_i, C_j) \rightarrow w_{ij}$ es una función que asocia un valor causal w_{ij} a cada par de neuronas (C_i, C_j) . Este valor w_{ij} denota la dirección e intensidad de la arista que conecta al concepto C_i con la neurona C_j donde la matriz de pesos W define el comportamiento del sistema (Kosko 1988). De forma similar, $A: (C_i) \rightarrow A_i$ es una función que asocia un grado de activación $A_i \in \mathbb{R}$ a cada concepto C_i del sistema, durante el tiempo t ($t = 1, 2, \dots, T$). Además, el mapa utiliza una función $f: \mathbb{R} \rightarrow [0, 1]$ para mantener el grado de activación de cada concepto en el rango adecuado.

1.2 PROCESO DE INFERENCIA DE LOS MAPAS COGNITIVOS DIFUSOS

Sin perder generalidad, el proceso de inferencia en un MCD se puede describir matemáticamente usando dos componentes: un vector de estado $A_{1 \times N}$ que representa el grado de activación de los conceptos del mapa, y una matriz de pesos causales $W_{N \times N}$ la cual define la interacción entre las neuronas. La siguiente ecuación (1.1) resume este proceso, el cual consiste en calcular el vector de estado A a través del tiempo, para una condición inicial A^0 . De forma análoga a otros sistemas neuronales, la activación de C_i dependerá de la activación las neuronas que inciden directamente sobre el concepto C_i y de los pesos causales asociados a dicho concepto.

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

Desde el punto de vista algebraico la fase de propagación de un estímulo se reduce a sucesivas multiplicaciones del vector de estado por la matriz de pesos, hasta que se alcance un criterio de parada. Por lo tanto, una variación en un concepto puede afectar directa o indirectamente a todo el mapa (Kosko 1988). Esto es esencial en escenarios de simulación, pues permite a los expertos estudiar el comportamiento del sistema para diferentes condiciones iniciales.

Dada su naturaleza recurrente, el sistema modelado por un MCD evolucionará durante el tiempo, donde la activación de cada neurona dependerá del grado de activación de sus antecedentes en la iteración anterior. Normalmente este proceso se repite hasta que el sistema estabilice o se alcance un número máximo de iteraciones (Kosko 1992). Los sistemas inestables pueden ser totalmente caóticos o cíclicos, y son frecuentes en modelos continuos. En resumen, el proceso de inferencia en un MCD puede mostrar una de las siguientes características:

- **Estados estables:** si $\exists t_k \in \mathbb{N} : A_i^{(t+1)} = A_i^{(t)}, \forall t > t_k$. En otras palabras, después de la iteración t_k el MCD producirá el mismo vector de estado. Desde luego, esta configuración es ideal, pues representa la codificación de un patrón oculto en la causalidad.
- **Estados cíclicos** (Wang et al. 1990): si $\exists t_k, P \in \mathbb{N} : A_i^{(t+P)} = A_i^{(t)}, \forall t > t_k$. El mapa tiene un comportamiento cíclico con periodo P . En este caso el sistema producirá el mismo vector de estado cada P -ciclos del proceso de inferencia.
- **Estados caóticos** (Wang et al. 1990): el mapa produce un vector de estado diferente en cada ciclo, o sea, los conceptos siempre varían su valor de activación.

En (Tsadiras 2008) se concluye que la convergencia de un MCD está estrechamente relacionada con la función de transformación utilizada para mantener el valor de activación de las neuronas en el intervalo adecuado. De acuerdo al tipo de salida, estas funciones pueden ser clasificadas en dos grandes categorías: discretas y continuas. Las funciones más frecuentemente utilizadas son: la función binaria, la función de tres estados y las variantes logísticas o sigmoideas (Bueno &

Salmeron 2009). Las dos primeras son discretas y nunca inducirán estados caóticos, sin embargo, esta premisa no se puede generalizar para las variantes sigmoideas.

Más explícitamente, los MCD son sistemas determinísticos y, por tanto, si el sistema produce un estado previamente visitado, entonces el mapa entrará en un ciclo cerrado. Por ejemplo, si se usa una función binaria el sistema producirá a lo sumo 2^N estados diferentes (asumiendo que N es el número de nodos del grafo). Esto significa que, en el peor escenario, el mapa alcanzará un estado visitado cada 2^N iteraciones. En el caso de la función de tres estados, el ciclo máximo tiene un periodo de 3^N , y por tanto el sistema nunca será caótico. Claramente esto no ocurre si se adoptan funciones de salida continua, donde el espacio de estados es un hipercubo.

Sin embargo, no es más factible utilizar funciones discretas, pues se ha demostrado que los MCD tienen mayor capacidad de inferencia si se usan funciones de transformación continuas (Tsadiras 2008). Incluso en (Bueno & Salmeron 2009) los autores demostraron que la variante sigmoidea tiene mayor capacidad predictiva, si se usa el mismo sistema de decisión. Independientemente de estos resultados, resulta conveniente destacar que la selección de la función frecuentemente está condicionada por los requerimientos del sistema, o sea, por rol que desempeñe cada concepto en el sistema modelado. En (Tsadiras 2008) se discuten varias reglas para la selección adecuada de este componente. Algunas de estas ideas son resumidas a continuación:

- **Los MCD binarios** (MCD con función de activación binaria) son apropiados para problemas altamente cualitativos, donde solamente se requiere representar el incremento o estabilidad de los conceptos que describen los estados del sistema.
- **Los MCD de tres estados** (MCD con función de activación de tres estados) son apropiados para problemas cualitativos, donde el experto puede representar el incremento, decremento o estabilidad de los conceptos que describen los estados del sistema.
- **Los MCD Sigmoideos** (MCD con función de transformación sigmoidea) son convenientes para modelar sistemas cualitativos y cuantitativos, donde es preciso modelar el decremento, incremento o estabilidad de los conceptos del sistema.

Otra característica que depende del sistema investigado está relacionada con la propagación de la información. Según la ecuación (1.1) un concepto recibe la influencia de las neuronas conectadas

directamente con él, pero no considera el hecho de que el propio concepto puede ser antecedente de su propia activación en cada iteración. La ecuación (1.2) soluciona este problema, confiriendo al mapa una capacidad memorística adicional. Sin embargo, esta variante en ocasiones conduce a neuronas sobresaturadas, limitando la capacidad para predecir nuevos patrones.

$$A_i^{(t+1)} = f \left(\sum_{j=1}^N w_{ji} A_j^{(t)} + A_i^{(t)} \right), i \neq j \quad (1.2)$$

Independientemente de la variante usada, el impacto que un concepto ejerce sobre otro se aprecia paulatinamente en cada iteración al calcular principalmente los efectos directos, ya que los nodos indirectos tienen un retardo (en número de iteraciones) igual a la cantidad de arcos que integren la ruta más corta. Los MCD tienden a obtener un valor único para el efecto total de los conceptos asociados directa e indirectamente a un concepto particular, a diferencia de los mapas causales que obtienen uno por cada neurona origen que tiene una relación directa o indirecta con él, por lo que se genera un vector de efectos en vez de un escalar.

1.3 ALGORITMOS DE APRENDIZAJE PARA MAPAS COGNITIVOS DIFUSOS

El objetivo principal de los algoritmos de aprendizajes para MCD es estimar una matriz de pesos que maximice el rendimiento del sistema. Este proceso es equivalente al ajuste de las conexiones causales entre los conceptos (similar a las relaciones sinápticas en una Red Neuronal Artificial). De forma general los algoritmos existentes se agrupan en tres grupos: los algoritmos Hebbianos, los enfoques poblacionales y los métodos de aprendizaje híbridos. (Nápoles 2014)

1.3.1 ALGORITMOS DE APRENDIZAJE HEBBIANOS

Los métodos Hebbianos son procedimientos no supervisados, y por tanto no requieren múltiples datos históricos etiquetados, donde se conoce el valor de los rasgos objetivos. El primer método de aprendizaje para MCD fue presentado por (Dickerson and Kosko, 1994) para ajustar los pesos causales. El algoritmo Hebbiano Diferencial (*Differential Hebbian Learning*, DHL) se basa en el siguiente principio: si el concepto causa C_i y el concepto efecto C_j cambian el valor de activación

de forma simultánea, entonces el peso causal w_{ij} se incrementará con un factor constante, en otro caso la relación de causalidad no será modificada en esa iteración.

La diferencia en el valor de activación ΔA_i del concepto i es: $\Delta A_i = A_i^{(t+1)} - A_i^{(t)}$. Este proceso se ejecuta iterativamente hasta que se obtenga una configuración deseada o se alcance un número máximo de iteraciones. La ecuación (1.3) resume el principio de este algoritmo de aprendizaje; 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 $[-1,1]$. Un criterio válido para seleccionar el valor del parámetro q puede ser el número máximo de iteraciones del algoritmo.

$$w_{ij}^{(t+1)} = \begin{cases} w_{ij}^{(t)} + \gamma(t) [\Delta A_i \Delta A_j - w_{ij}^{(t)}] & \text{si } \Delta A_i \neq 0 \\ w_{ij}^{(t)} & \text{si } \Delta A_i = 0 \end{cases} \quad (1.3)$$

La principal desventaja de este enfoque local radica en la ausencia de información concerniente al sistema como un todo. En otras palabras, DHL modifica los pesos teniendo en cuenta cambios en el valor de activación entre los pares de conceptos, ignorando la influencia proveniente de las demás neuronas. Para atenuar este problema, (Hueriga 2002) introdujo el Algoritmo Diferencial Balanceado (*Balanced Differential Algorithm*, BDA). La idea central consiste en usar el grado de activación de los nodos que se modifican simultáneamente, durante el proceso de actualización de la matriz de pesos. La siguiente ecuación (1.4) formaliza este método, asumiendo un MCD sin conceptos con auto-conexiones, el cual se ajusta de forma iterativa.

$$w_{ji}^{(t+1)} = \begin{cases} w_{ji}^{(t)} + \gamma(t) \left[\frac{\Delta A_i / \Delta A_j}{\sum \Delta A_i / \Delta A_k} - w_{ji}^{(t)} \right] & \text{si } \Delta A_i \Delta A_j > 0, \forall C_k: \Delta A_i \Delta A_k > 0 \\ w_{ji}^{(t)} + \gamma(t) \left[\frac{-\Delta A_i / \Delta A_j}{\sum \Delta A_i / \Delta A_k} - w_{ji}^{(t)} \right] & \text{si } \Delta A_i \Delta A_j < 0, \forall C_k: \Delta A_i \Delta A_k < 0 \end{cases} \quad (1.4)$$

Si hay un cambio en el valor de activación del nodo C_j en la misma dirección que el concepto C_i ($\Delta A_i \Delta A_j > 0$), entonces w_{ji} se suma como una relación positiva, en caso contrario ($\Delta A_i \Delta A_j < 0$) se asume una relación de causalidad negativa. En este algoritmo, el peso w_{ji} se ajusta a través del cociente $\Delta A_i / \Delta A_j$. Este valor representa una medida del factor de cambio del concepto C_i cuando el concepto C_j cambia durante el tiempo. Note que la normalización del factor permite calcular el

ajuste de la causalidad teniendo en cuenta el valor relativo del cambio de la neurona C_j , respecto al valor de activación de los demás conceptos que influyen al concepto C_i . El algoritmo BDA es más poderoso que su predecesor, aunque no es muy popular (hasta donde se conoce solo se ha aplicado en problemas de reconocimiento de patrones binarios).

En otro intento por ajustar la matriz de pesos causales (Papageorgiou et al. 2004) introdujeron el Algoritmo Hebbiano Activo (*Active Hebbian Learning*, AHL). Los algoritmos anteriores asumen que en cada iteración las neuronas se actualizan de forma síncrona, sin embargo, AHL considera que los nodos se activan de forma asíncrona, siguiendo una secuencia. A diferencia de los demás algoritmos, AHL modifica todos los pesos casuales en cada iteración (exceptuando los pesos de la diagonal principal), aunque el experto debe especificar tres elementos:

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

Si estas especificaciones son conocidas, entonces AHL es capaz de estimar la matriz de pesos a través de un procedimiento iterativo de siete pasos (Papageorgiou et al. 2004). La principal desventaja de este método es su dependencia del criterio de expertos.

Adicionalmente, en (Papageorgiou et al. 2003) los autores presentan un Aprendizaje Hebbiano No-Lineal (*Nonlinear Hebbian Learning*, NHL), el cual es una extensión no lineal de ley Hebbiana. La idea central del método es modificar los pesos que fueron inicialmente especificados por los expertos, hasta que satisfaga un criterio de parada. En este esquema el signo de cada conexión corresponde con la interpretación física del sistema y constituye una restricción del modelo. De forma general el experto del dominio tiene que especificar los siguientes aspectos:

- Definición 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.

En otras palabras, el algoritmo debe respetar la dirección de la causalidad así que sólo se ajustará la magnitud de la interacción entre las neuronas. La ecuación (1.5) formaliza cómo se ajustan los pesos en este modelo, donde γ es un factor que denota la disminución del peso, η es la velocidad

del aprendizaje y $sgn(\cdot)$ es la función signo. Adicionalmente el algoritmo introduce dos criterios de paradas novedosos. En el primer caso se finaliza cuando se alcance una solución próxima a la respuesta deseada, mientras que el segundo caso el algoritmo finaliza cuando se detecte un punto atractor de equilibrio. NHL es un algoritmo muy eficiente debido a su simplicidad, y su principal desventaja radica en la dependencia al criterio inicial de los expertos.

$$w_{ij}^{(t+1)} = \gamma w_{ij}^{(t)} + \eta A_j^{(t)} \left(A_i^{(t)} - sgn(w_{ij}^{(t)}) A_j^{(t)} w_{ij}^{(t)} \right) \quad (1.5)$$

De forma general los algoritmos Hebbianos son procedimientos no supervisados diseñados para ajustar la matriz de pesos causales y que utilizan un único ejemplo en su esquema de aprendizaje. Sin embargo, el objetivo de estos métodos no es maximizar la capacidad del mapa para predecir nuevos patrones (ej. exactitud de la clasificación) sino lograr que mapa converja a un punto fijo y que la salida corresponda con un escenario determinado. Este esquema se enfoca principalmente en problemas de simulación. Recientemente se han publicado algunos trabajos que exploran una nueva arista de estas estructuras neuronales: la clasificación de patrones. Para este contexto estos algoritmos de aprendizaje no resultan adecuados, pues no es suficiente asegurar la convergencia del sistema a un atractor, también debe considerarse la exactitud.

En un intento de aliviar el problema de la precisión del mapa (Stach et al. 2008) propusieron una variante del método NHL. El nuevo modelo, llamado NHL Guiado por Datos (*Data-driven NHL*, DDNHL), utiliza el mismo principio de aprendizaje que su predecesor, y conjuntamente utiliza datos históricos para mejorar la calidad del sistema resultante. En (Stach et al. 2008) los autores demostraron a través de varios experimentos que DDNHL es superior a NHL si existen los datos históricos, pero en un estudio más exhaustivo (Papakostas et al. 2012) se concluyó que incluso esta variante tiene un desempeño pobre en problemas de clasificación.

1.3.2 ALGORITMOS DE APRENDIZAJE BASADOS EN METAHEURÍSTICAS

En literatura se han propuesto varios algoritmos de aprendizaje para computar la matriz de pesos de un MCD utilizando metaheurísticas poblacionales. La idea central de este enfoque consiste en generar soluciones que minimicen una función de error, o sea, las diferencias entre las respuestas

conocidas y las salidas inferidas por mapa. Estos métodos son más costosos computacionalmente pues requieren optimizar una función objetivo, y adicionalmente requieren múltiples secuencias de entrada para el proceso de entrenamiento. Una secuencia usualmente codifica la activación de los conceptos de entrada del mapa y la respuesta conocida para ese estímulo.

En este esquema cada individuo denota una matriz de pesos candidata, por tanto, cada punto en el espacio de búsqueda representará un conjunto de pesos causales. Por tanto, un sistema definido por N nodos se codificará en un vector con dimensión a lo sumo $(N \times N)$, pues las relaciones causales con peso cero no son modificadas. Más concretamente, cada dimensión del individuo se asocia con una conexión causal. La Figura 1 ilustra la idea general de esta codificación para un mapa parcialmente conexo que involucra seis conceptos de entrada.

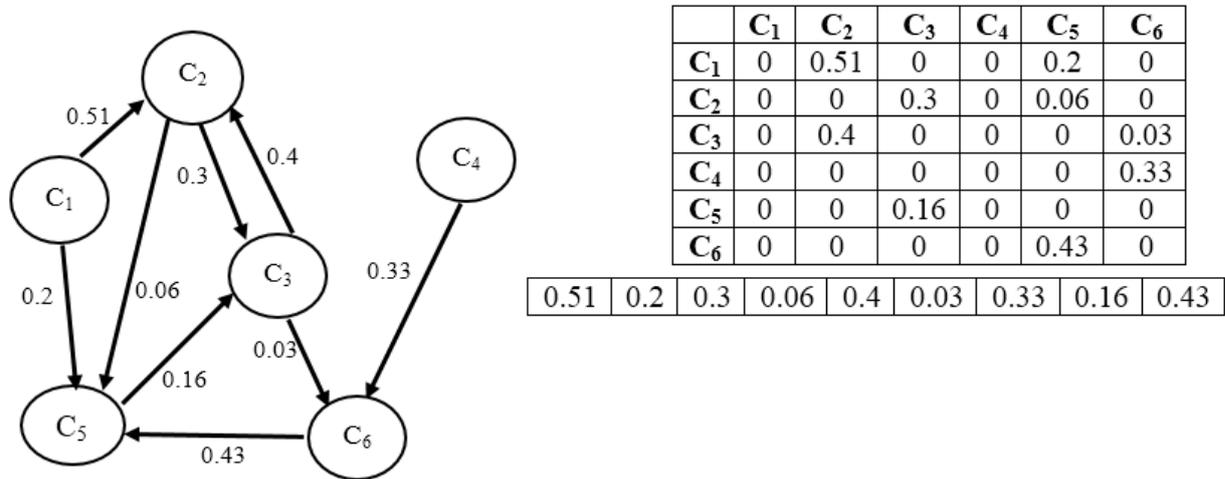


Figura 1. Codificación de un individuo en un algoritmo de aprendizaje poblacional.

Inspirados en este enfoque, (Koulouriotis et al. 2001) utilizaron un método basado en Estrategia Evolutiva para ajustar la estructura del mapa. Este algoritmo utiliza datos históricos, donde cada instancia está compuesta por un par de vectores. El primer vector codifica el grado de activación de los conceptos de entrada, mientras que el segundo vector denota la respuesta del sistema para la configuración especificada. La ecuación (1.6) formaliza la función objetivo asociada, donde K es el número total de instancias, A_i denota la respuesta actual del mapa, y \tilde{A}_i es la salida esperada para el i -ésimo concepto. A diferencia de los métodos Hebbianos, este modelo requiere múltiples secuencias de entrada para estimar las relaciones causales del sistema.

$$f(x) = \sum_{k=1}^K \sum_{i=1}^N |A_i - \tilde{A}_i| \quad (1.6)$$

Siguiendo un lógica similar, (Parsopoulos et al. 2003) aplicaron Optimización de Enjambres de Partículas (*Particle Swarm Optimization*, PSO) para estimar la estructura adecuada del sistema a partir de una secuencia de múltiples vectores de estado. La peculiaridad de este método radica en su esquema de aprendizaje: la respuesta del sistema la definen nodos decisores. Un experto debe establecer los límites de activación A_i^{\min} y A_i^{\max} permitidos para cada nodo decisión. Durante el progreso del algoritmo se explora el espacio de búsqueda con el objetivo de encontrar una matriz de pesos que satisfaga las restricciones impuestas por el experto. La siguiente ecuación formaliza la función objetivo asociada, donde H representa la función *Heaviside* (Parsopoulos et al. 2003), mientras que A_i^{DC} denota el i -ésimo concepto decisor del sistema.

$$f(x) = \sum_{i=1}^N H(A_i^{\min} - A_i^{DC}) |A_i^{\min} - A_i^{DC}| + \sum_{i=1}^N H(A_i^{DC} - A_i^{\max}) |A_i^{DC} - A_i^{\max}| \quad (1.7)$$

En (Stach et al. 2005) los autores propusieron un enfoque automatizado basado en Algoritmos Genéticos con Codificación Real (*Real-Coded Genetic Algorithm*, RCGA). La idea del método consiste en maximizar la función objetivo $f(x) = 1/(1 + \alpha * E(x))$, reduciendo el error global del mapa (véase ecuación 1.8). En esta ecuación α denota el coeficiente de calidad del algoritmo, $A_i(t)$ es la respuesta conocida en el tiempo t , mientras que $\tilde{A}_i(t)$ denota la respuesta del sistema usando la matriz de pesos candidata, para un estímulo inicial. Este método es capaz de computar mapas con alta capacidad de predicción, sin embargo, su rendimiento decrece significativamente cuando el número de variables en el sistema aumenta ligeramente.

$$E(x) = \frac{1}{N(T-1)} \sum_{t=1}^{T-1} \sum_{i=1}^N |A_i(t) - \tilde{A}_i(t)| \quad (1.8)$$

En este procedimiento la intervención de expertos no se requiere, de hecho, este método es capaz de estimar una configuración adecuada utilizando una única instancia, aunque también se pueden utilizar múltiples secuencias de entrada a través del tiempo.

Más recientemente, en (Chen et al. 2012) los autores propusieron un novedoso método inspirado en Optimización de Colonias de Hormigas (*Ant Colony Optimization*, ACO) para ajustar mapas

con docenas de conceptos. En este modelo cada peso se codifica como una secuencia de estados discretos, o sea, la parte entera y las cifras de precisión decimal. Durante el proceso de búsqueda, cada hormiga construye una solución. En el próximo paso, cada solución discreta se transforma a su equivalente continua, resultando en una matriz de pesos causales. La ecuación (1.9) formaliza la función de evaluación que se minimiza en este procedimiento.

$$E(x) = \frac{1}{NK(T-1)} \sum_{k=1}^K \sum_{i=1}^N \sum_{t=1}^{T-1} (A_i^k(t) - \tilde{A}_i^k(t))^2 \quad (1.9)$$

Note que estos métodos no pueden garantizar la estabilidad de los mapas utilizados en problemas de clasificación de patrones. Más explícitamente, en estos problemas solo se conoce la respuesta esperada para un vector de estímulo inicial. Esto significa que el objetivo del algoritmo se reduce a minimizar las diferencias entre las respuestas del mapa y las salidas esperadas, sin considerar la variabilidad del sistema durante el proceso de inferencia. Como consecuencia se obtienen mapas con alta capacidad de predicción, pero frecuentemente inestables.

1.4 MAPAS COGNITIVOS DIFUSOS PARA MODELACIÓN Y CLASIFICACIÓN

Los MCD son un motor de inferencia eficiente para el modelado de relaciones causales complejas con facilidad, tanto cualitativa como cuantitativamente. En los últimos años, los MCD también se han asociado con la tarea de clasificación. Desde una perspectiva de aprendizaje de máquina, pueden ser vistos como redes neuronales sin capas ocultas. Según su uso, pueden separarse en MCD para modelación y MCD para clasificación, como se muestra en la Figura 2.

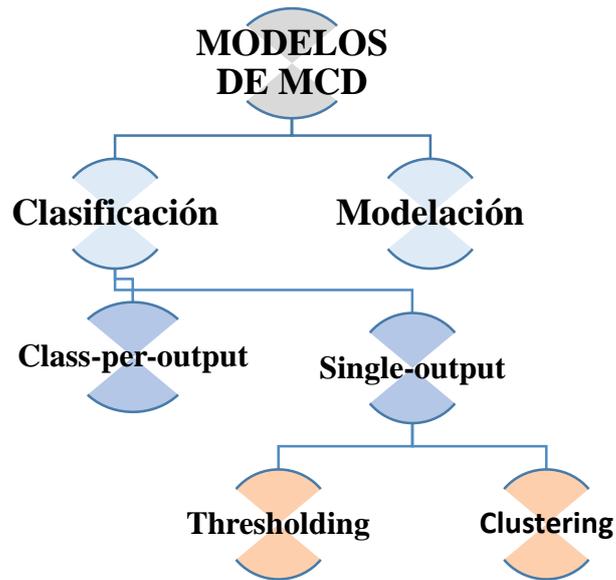


Figura 2. Modelos de Mapas Cognitivos Difusos

1.4.1 MODELACIÓN

La fortaleza de los MCD está relacionada principalmente con propósitos de simulación y análisis de escenarios. En este tipo de MCD, todas las neuronas denotan variables del sistema físico bajo análisis. No existen neuronas de entrada o de salida, ni hay clases de decisión asociadas al modelo (aunque pueden ser inferidas de los estados del sistema).

El grado de activación de una neurona es un indicador del nivel de presencia del concepto en el sistema (Kosko 1992). Esta característica es un aspecto clave en la interpretabilidad y usabilidad de los MCD, sobre todo si la función de umbral es continua (dada la importancia de esta función para la presente propuesta sus propiedades serán analizadas más adelante). La Figura 3 muestra la apariencia de un MCD y la semántica en la activación de una neurona.

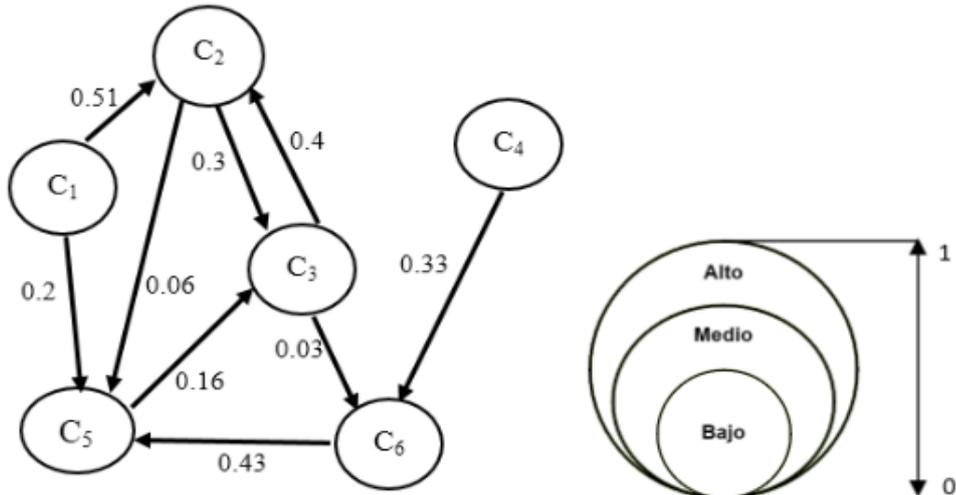


Figura 3. Ejemplo de MCD y semántica de activación de una neurona.

El carácter difuso de los MCD se observa en la construcción del mapa, ya que la Lógica Difusa no se aplica en el proceso de propagación de la información. Los conceptos y relaciones del mapa son representados por variables difusas expresadas como términos lingüísticos. Más explícitamente, durante la etapa de ingeniería del conocimiento cada experto expresa la relación que existe entre cada par de conceptos C_i y C_j del mapa. Entonces, para cada relación causal se obtienen k reglas con la siguiente estructura: **SI** C_i es A **ENTONCES** C_j es B y el peso w_{ij} es C.

Luego se utiliza el método del centroide y el mecanismo de inferencia Mamdani para agregar las k reglas, y el valor defuzificado es el valor de la relación. Por ejemplo, suponga que los términos lingüísticos asociados a la causalidad son: *muy bajo*, *bajo*, *medio*, *alto* y *muy alto*. La Figura 4 muestra las funciones de pertenencia para dos expertos del dominio. El primer experto considera que si el concepto origen C_i se activa totalmente, entonces el concepto efecto C_j tendrá un grado de activación alto con certeza 0.6. Por su parte, el segundo experto considera que bajo la misma premisa, el nivel de activación del concepto C_j será muy alto con certeza 0.3. Esto implica que el valor defuzificado para el peso w_{ij} será 0.76. De forma similar se estima el resto de la matriz de pesos, para todas las conexiones causales que intervienen en el modelo. (Nápoles 2014)

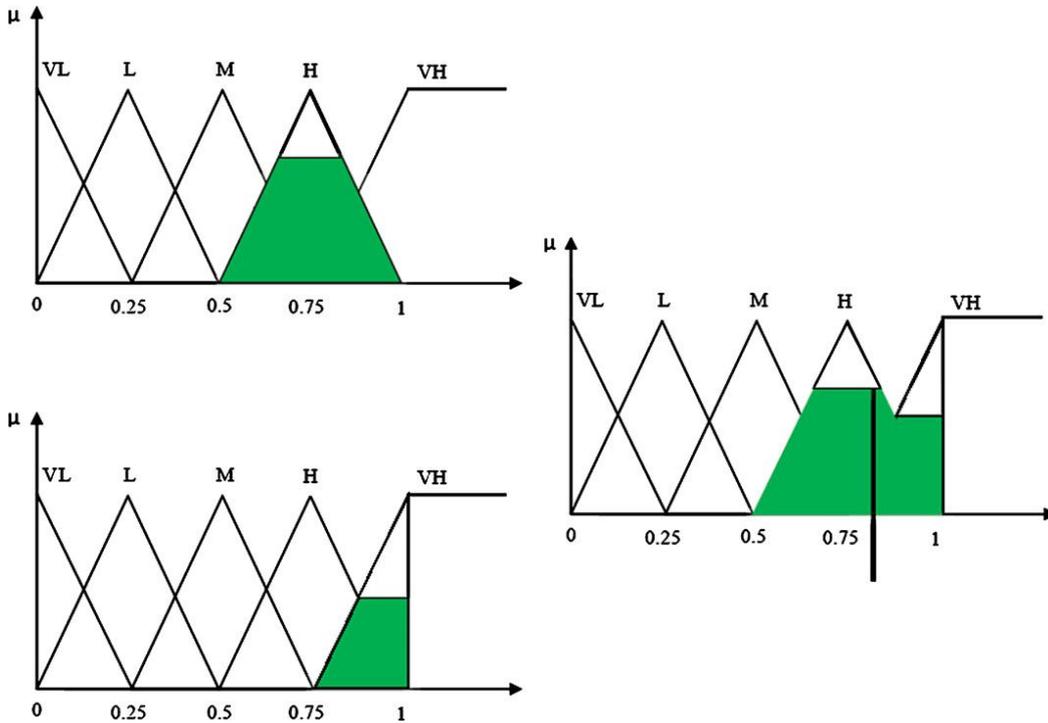


Figura 4. Proceso de construcción de un MCD donde intervienen dos expertos del dominio. El primer experto considera que el efecto de C_i sobre C_j es alto, mientras que el segundo considera que es muy alto.

De acuerdo a (Codara 1998), los MCD pueden ser usados para varios propósitos, incluyendo cuatro funciones:

1. Explicativa: Se centra en la reconstrucción de las premisas detrás del comportamiento de agentes dados, la comprensión de las razones para sus decisiones y de las acciones que realizan y el resaltado de las distorsiones y límites en su representación de la situación.
2. Predicción: Esta función se basa en la predicción de las decisiones y acciones futuras, o las razones que un agente dado usará para justificar las nuevas ocurrencias.
3. Reflectante: Esta función ayuda a quienes toman las decisiones para reflexionar sobre su representación de una situación determinada, con el fin de determinar su idoneidad y posiblemente provocar la introducción de los cambios necesarios.
4. Estratégica: Se basa en la generación de una descripción más precisa de una situación compleja.

En términos generales, un MCD se construye mediante la mezcla de la experiencia y los conocimientos en relación con un sistema. Esto se puede lograr mediante un equipo de expertos

humanos para describir la estructura y comportamiento del sistema en diferentes condiciones. MCD son una forma sencilla para encontrar qué factor debe modificarse y cómo hacerlo. En este sentido, un MCD es una técnica de modelado dinámico en la que la resolución de la representación del sistema puede aumentar aún más mediante el mapeo. Las principales motivaciones para el uso del enfoque de MCD son los siguientes: fácil de construir y parametrizar, flexibilidad en la representación, fácil de usar, fácil de entender, ejecución rápida, capaz de manejar problemas complejos relacionados con adquisición de conocimientos y gestión, y manejar los efectos dinámicos debidos a la estructura de retroalimentación del sistema modelado (Papageorgiou & Salmeron 2013).

1.4.2 CLASIFICACIÓN

En el caso de los MCD para clasificación, todos los conceptos pueden agruparse en *conceptos de entrada* o *conceptos de salida*, a la vez que debe especificarse un modelo de decisión para definir un protocolo de clasificación. Un primer intento para incorporar los MCD a tareas de reconocimiento de patrones fue implementado por (Papakostas et al. 2008).

Los modelos mostrados a continuación son de una neurona de salida por cada clase de decisión (*class-per-output*). La Figura 5 adopta una relación causal entre las clases de decisión y características sin correlación. El modelo de la Figura 6 supone una relación causal entre las características, pero con clases no correlacionadas. La Figura 7 solo adopta relaciones causales entre las características y las clases de decisión. En la Figura 8 todas las características están provistas con un modelo de *caja negra* que realiza un primer proceso de clasificación. Luego, los valores de decisión se asignan a los valores de activación en la capa de entrada, para ser vueltos a visitar por la capa de salida, corrigiendo así la clasificación final. Similarmente al modelo de la Figura 9, todas las características son pre-procesadas por un clasificador de *caja negra*. Además, los conceptos de entrada están relacionados directamente con los de salida.

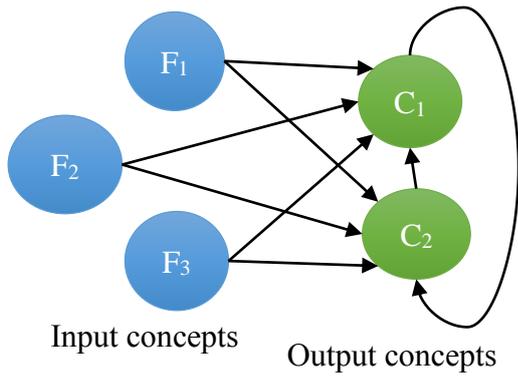


Figura 5. Ejemplo de estructura para clasificador basado en MCD.

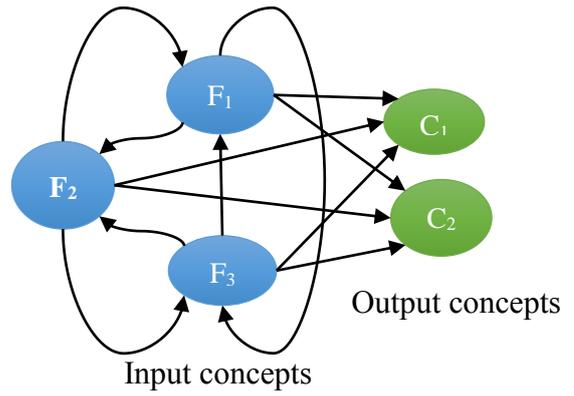


Figura 6. Ejemplo de estructura para clasificador basado en MCD.

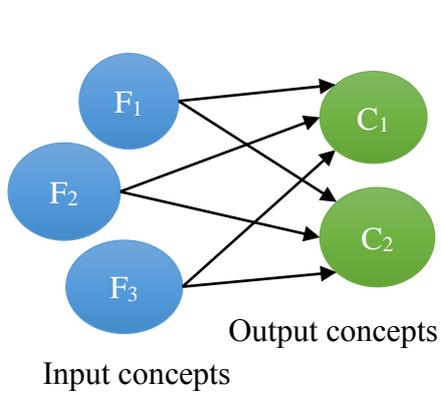


Figura 7. Ejemplo de estructura para clasificador basado en MCD.

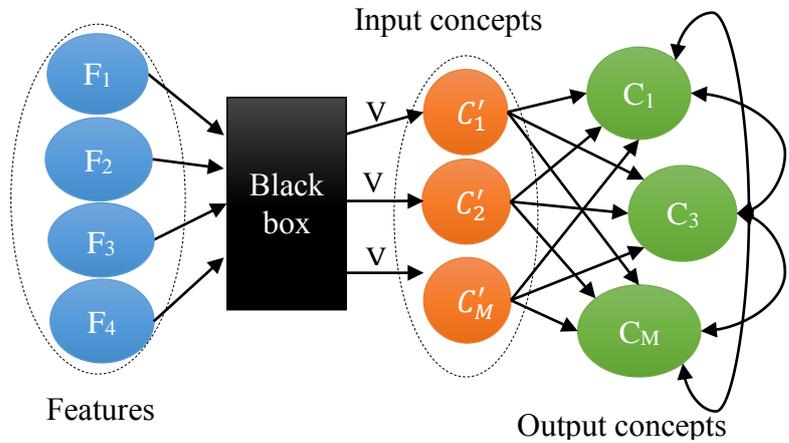


Figura 8. Ejemplo de estructura para clasificador basado en MCD.

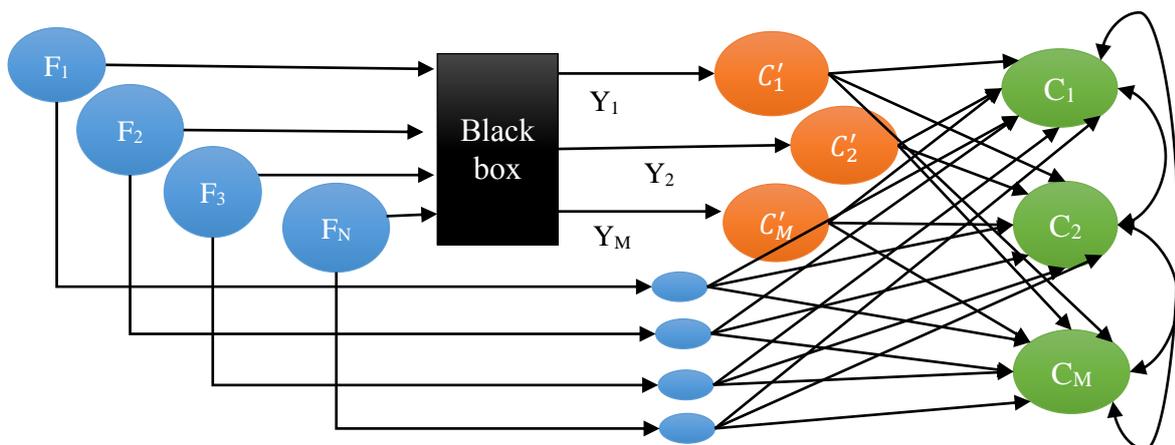


Figura 9. Ejemplo de estructura para clasificador basado en MCD.

Otros tipos de modelos para la clasificación son los de salida única (*single-output*) por umbralización (*thresholding*) y por agrupamiento (*clustering*). En los de umbralización, siendo usualmente usados en procedimientos de aprendizaje no supervisado, se extraen umbrales de salida específicos que separan las clases del problema entre ellas. El proceso de extracción de umbrales puede ser difícil porque el método puede ser superado mediante la aplicación de un procedimiento más confiable que consiste en el agrupamiento de los valores de los conceptos de salida (clase con distancia mínima) y la definición de los centros de los grupos durante la fase de entrenamiento (Papakostas et al. 2012). Existen dos formas de hacer esto, antes del paso de aprendizaje:

- Agrupar todos los valores de salida, determinar los centroides y etiquetar los grupos que emergen (Figura 10).
- Determinar los umbrales para cada clase usando instancias de entrenamiento en el proceso de aprendizaje (Figura 11).

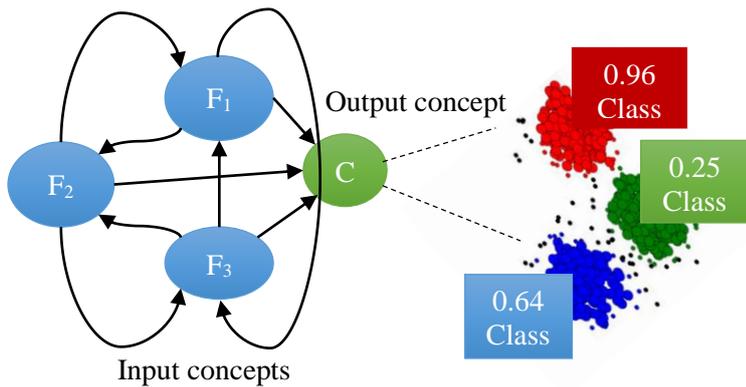


Figura 10. Modelo Single-output (clustering).

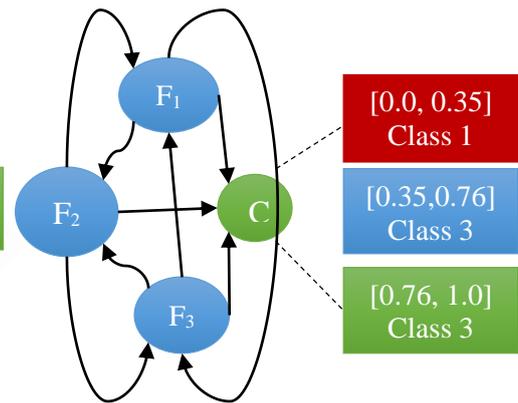


Figura 11. Modelo Single-output (thresholding).

Desde su inicio, los MCD jugaron un rol vital en las aplicaciones de áreas científicas diversas, pero es aún muy cuestionado su uso en clasificación de patrones. Este es un campo casi sin explorar en la literatura, sobre el cual existen pocos experimentos, pero abre el camino a nuevas investigaciones. A pesar de esto, muchas aplicaciones se han llevado a cabo en este campo, obteniéndose resultados convincentes y mejores capacidades de clasificación respecto a otros clasificadores.

1.5 ESTABILIDAD EN MAPAS COGNITIVOS DIFUSOS

Los algoritmos de aprendizaje ilustrados en la sección 1.3 no consideran la convergencia del mapa final, y esta observación es extensiva a otras propuestas. En otras palabras, la mayoría de los métodos de aprendizaje para MCD tienen como objetivo la estimación de los pesos causales, ignorando las propiedades de estabilidad del sistema final. En la literatura se han reportado pocos estudios analíticos sobre la convergencia de los MCD, principalmente debido a la complejidad del problema matemático asociado. En esta sección se abordan los resultados más prometedores en esta dirección, y se explica por qué no son aplicables en todos los casos. (Nápoles 2014)

En (Boutalis et al. 2009) se analizó la existencia y unicidad de los valores de equilibrio para los conceptos de los MCD. Esto es teniendo en cuenta funciones de transformación sigmoideas con propiedades contractivas o, al menos, no expansivas. En (Boutalis et al. 2009) se presentaron dos teoremas (ver Teoremas 1 y 2) sobre la estabilidad de MCD Sigmoideos. La demostración analítica de estos teoremas se basa en las propiedades de las contracciones en un espacio métrico completo (Rudin 1964).

Teorema 1. Existe una y solo una solución para cualquier valor de activación A_i de cualquier MCD con función de transformación $f = \frac{1}{1+e^{-x}}$, si es cierta la siguiente expresión:

$$\left(\sum_{i=1}^N \|w_i\| \right)^{1/2} < 4 \quad (1.10)$$

Donde w_i es la i -ésima fila de la matriz de pesos del mapa y $\| \cdot \|$ es la norma l_2 . Si se cumple la igualdad, entonces existe al menos una salida, no necesariamente única, para cada vector de entrada A_i .

Para comprender las limitaciones del teorema anterior, suponga un problema de simulación de escenarios donde se tengan k escenarios (k vectores de activación iniciales y para cada uno el sistema debe dar un vector salida). Si la matriz del MCD cumple la condición (1.10) entonces, sea cual sea el vector de activación inicial, el sistema siempre llegará al mismo vector final. Así, todo escenario lleva al mismo estado atractor de equilibrio; por lo que quedan fuera de consideración, tanto el estímulo inicial, como los valores asociados a las relaciones causales del sistema.

Otra limitación es la relacionada con la exactitud. El sistema no tiene capacidad de predicción, ya que todo escenario lleva al mismo estado atractor de equilibrio. Los únicos contextos posibles para la aplicación de este teorema son cuando se emplea un único estímulo en la simulación o cuando se desea la misma salida para cualquier estímulo inicial (ej. sistemas de control).

Por su parte, el Teorema 2 es levemente diferente pues asume un MCD Sigmoido con neuronas de entrada¹. En este caso, el estado de equilibrio dependerá de las características de la matriz de pesos y del vector de estímulo. En otras palabras, el teorema asegura que existirá un único punto fijo para cada instancia del problema. Además, en (Boutalis et al. 2009) se presenta un algoritmo de aprendizaje basado en el método del gradiente para calcular una matriz de peso capaz de producir una salida acorde a la respuesta conocida, y que no comprometa la estabilidad del sistema. A pesar de ser un resultado muy prometedor, tampoco es aplicable a problemas de predicción ni de simulación de varios escenarios.

Teorema 2. Para un MCD Sigmoido con m conceptos de entrada, existe una y solo una solución para cualquier valor de activación A_i si es cierta la siguiente expresión:

$$\left(\sum_{i=1}^N \|w_{m+i}\| \right)^{1/2} < 4 \quad (1.11)$$

Donde w_{N+i} es la $(N + i)$ -ésima fila de la matriz de pesos del mapa y $\| \cdot \|$ es la norma l_2 . Si se cumple la igualdad, entonces existe al menos una salida, no necesariamente única, para cada vector de entrada A_i .

En otras palabras, el algoritmo de adaptación de los pesos estima una matriz de pesos estable que satisface la respuesta del sistema esperada, para una única instancia del problema. Si el problema tiene múltiples instancias (como es típico en problemas de aprendizaje automatizado) entonces el algoritmo debe ser aplicado sucesivamente. Como resultado se obtiene un conjunto diferente de pesos para cada instancia del problema, y por lo tanto no hay aprendizaje. (Nápoles 2014)

¹ Los autores definen los conceptos de entrada como neuronas que influyen a las demás, pero ellas no son influenciadas por las demás neuronas. En otras palabras, el valor de activación de estos conceptos permanece invariable durante el proceso de inferencia del mapa.

En (Kottas et al. 2012) se extiende el resultado del *Teorema 1* permitiendo que cada neurona tenga su propia función de transformación sigmoidea.

Teorema 3. Existe una y solo una solución para cualquier valor de activación A_i de cualquier MCD con función de transformación $f_i = \frac{1}{1+e^{-\lambda_i x}}$ si es cierta la siguiente expresión:

$$\left(\sum_{i=1}^M (\lambda_i \ell_i \|w_i\|)^2 \right)^{1/2} < 1 \quad (1.12)$$

Donde w_i es la i -ésima fila de la matriz de pesos del mapa, $\| \cdot \|$ es la norma l_2 , ℓ_i es la derivada parcial $\partial f_i / \partial x$ y f_i es la función asociada a la neurona correspondiente. Si se cumple la igualdad, entonces existe al menos una salida, no necesariamente única, para cada vector de entrada A_i .

Otro resultado teórico fue publicado en (Knight, Lloyd & Penn 2014) donde los autores demuestran que el número de atractores de un MCD Sigmoideo está determinado por la inclinación de la función de umbral (Teorema 4). Desde el punto de vista analítico este teorema es una ligera variación de los Teoremas 1, 2 y 3, pero enfocado a las propiedades que debe satisfacer la función de normalización para que el mapa sea linealmente estable. Según los autores, asegurar la existencia y unicidad del punto fijo en un MCD conduce a sistemas más confiables, incluso cuando el grado de activación de las neuronas no es cuantificado de forma precisa por su carácter difuso. (Nápoles 2014)

Teorema 4. El número de soluciones de la regla de inferencia neuronal con $f = \frac{1}{1+e^{-\lambda(x-h)}}$ depende de la inclinación λ de la función.

- a) Si $\lambda \geq 0$ es *suficientemente pequeño* entonces el mapa tendrá una única solución, y esta solución será linealmente estable.
- b) Si $\lambda \geq 0$ es *suficientemente grande* entonces el mapa tendrá múltiples soluciones, y muchas de estas soluciones pueden ser linealmente estables.

Los autores también determinan analíticamente las cotas superior e inferior para el parámetro λ , permitiendo la definición de los términos *suficientemente pequeño* y *suficientemente grande*, respectivamente. Ambos términos son de naturaleza difusa.

Al igual que los otros teoremas este resultado no es aplicable a MCD utilizados en problemas de clasificación de patrones o toma de decisiones (Nápoles, Bello, et al. 2014), ni tampoco en problemas de simulación con varios escenarios. A continuación, se muestran ventajas y limitaciones de los resultados analíticos mostrados previamente.

Ventajas:

- Los resultados teóricos anteriores son especialmente convenientes en ambientes de control cuando el MCD debe producir la misma respuesta sin importar el vector de activación inicial.
- Se conocen las condiciones exactas que necesitan satisfacerse para asegurar la convergencia del sistema sin usar complejos procesos de optimización.
- Permiten entender los fundamentos básicos de los MCD. La falta de resultados analíticos es la limitante principal para hibridar mapas cognitivos con otras teorías bien establecidas (ej. Redes Neuronales).

Limitaciones:

- La limitante principal es que se obtienen MCD sin capacidades de clasificación.
- Por consiguiente, estos teoremas no sirven para problemas de clasificación, ya que la unicidad del punto atractor de equilibrio llevan a la misma clase de decisión una y otra vez. Esto ocurre porque los clasificadores convergen al mismo atractor de punto fijo, sin importar las características intrínsecas de los patrones de entradas de las relaciones causales.
- Algunos de estos resultados analíticos imponen condiciones específicas sobre la matriz de pesos, pero los pesos causales son frecuentemente fijados por expertos.
- En principio, no podemos forzar al experto del dominio para proveer un conjunto de pesos causales con características de convergencia mejoradas. De otro modo, el conocimiento comprimido dentro del modelo estaría basado en la convergencia del sistema y no en las relaciones causales entre las variables del problema.

En resumen, el principal problema de los teoremas discutidos en esta sección no está relacionado con la unicidad del atractor de equilibrio, sino con los criterios de convergencia utilizados en las

demostraciones. Estos criterios aseguran que el atractor existe y es único, pero resulta el mismo estado de equilibrio para todos los objetos del problema, lo cual es un enfoque incorrecto en problemas de simulación de escenarios.

1.6 CONSIDERACIONES FINALES DEL CAPÍTULO

Los MCD son una técnica basada en el conocimiento que permite el razonamiento causal y la simulación de complejos sistemas dinámicos. Usualmente ellos son construidos de forma manual por expertos o de forma automática. Durante la etapa de ingeniería del conocimiento un aspecto central es la estimación de la dirección e intensidad de las relaciones causales entre las variables del modelo, lo cual determina la efectividad y eficiencia de la modelación. Para lidiar con este problema se han desarrollado varios algoritmos de aprendizajes:

- No supervisados (ej. algoritmos Hebbianos).
- Supervisados (ej. inspirados en metaheurísticas).

A pesar de su éxito relativo, los algoritmos de aprendizaje existentes no consideran cuestiones de estabilidad como parte de su esquema, solo incluyen aspectos relacionados con la precisión final del proceso de inferencia. Como consecuencia se obtienen sistemas con capacidad de predicción comparable con los clasificadores tradicionales (ej. Redes Bayesianas, Árboles de Decisión) pero esta precisión es engañosa si el sistema no es estable. Sin embargo, garantizar la convergencia en un MCD no es trivial, sobre todo en mapas que utilizan funciones continuas donde el dominio de excitación de las neuronas está definido sobre un hipercubo difuso.

Por otra parte, los métodos analíticos que garantizan la existencia y unicidad del punto atractor de equilibrio no pueden ser aplicados en contextos donde se requiere la simulación de múltiples secuencias de entrada, sino en problemas donde existe un único estímulo. Estos métodos limitan significativamente la capacidad de inferencia de los MCD pues no consideran los ejemplos en su esquema de convergencia. De forma general, el problema de la estabilidad en MCD Sigmoideos persiste como un problema para los matemáticos y la ausencia de una solución exacta evidencia la necesidad de enfoques alternativos inspirados en técnicas aproximadas.

CAPÍTULO 2

ALGORITMO PARA MEJORAR LA
CONVERGENCIA EN MAPAS
COGNITIVOS DIFUSOS SIGMOIDEOS

CAPÍTULO 2. ALGORITMO PARA MEJORAR LA CONVERGENCIA EN MAPAS COGNITIVOS DIFUSOS SIGMOIDEOS

En este capítulo se introduce un nuevo algoritmo de aprendizaje para mejorar las propiedades de convergencia en MCD Sigmoideos, preservando la precisión, una vez que se ha estimado la causalidad. El método usa una función sigmoidea para cada neurona del mapa, en lugar de usar la misma función de umbral para todos los conceptos. En la práctica, el procedimiento se reduce a la estimación de la inclinación y el desplazamiento adecuados para cada función sigmoidea, lo cual conduce a un problema de optimización continuo que puede ser solucionado usando diversas estrategias. De esta forma es posible obtener mapas más estables, preservando la precisión y sin modificar la matriz de pesos del sistema.

2.1 PRELIMINARES SOBRE EL ALGORITMO

En el capítulo anterior se revisaron varios métodos analíticos que permiten encontrar condiciones que garanticen la estabilidad en los MCD Sigmoideos. El principal problema de estos modelos está relacionado con la reducción de la capacidad de predicción del sistema. Más explícitamente, después de aplicar estos algoritmos, el mapa es capaz de converger a un estado de equilibrio que es único para todas las instancias del problema, sin importar la naturaleza del vector de estímulo codificado en los casos de entrada. Esto implica que para cada escenario se obtendrá la misma salida, por lo que solo se preserva la modelación del sistema.

Un enfoque adecuado para asegurar la estabilidad en un MCD Sigmoideo sin afectar su precisión debe considerar los estímulos iniciales y los valores de activación finales esperados para dichos estímulos. De esta forma el mapa convergerá al punto de equilibrio (preferiblemente único) correspondiente a cada estímulo.

El Teorema 3, descrito en el capítulo anterior, permite afirmar que la inclinación de la función sigmoidea tiene relación con la estabilidad del mapa. Basados en esta hipótesis, se diseña un experimento donde se varía aleatoriamente la inclinación de la función de transformación. La

siguiente ecuación muestra la formulación de la función de transformación asociada a cada concepto C_i del mapa.

$$f_i(x) = \frac{1}{1 + e^{-\lambda_i(x-h_i)}} \quad (2.1)$$

En (Nápoles, Bello, et al. 2014) se busca el impacto real de la variación de λ_i en (2.1) pero con $h_i = 0.5$, concluyéndose que pequeños cambios en λ_i afectan la estabilidad del mapa. Concretamente este estudio mostró que la variabilidad en las respuestas de los mapas inestables ocasionalmente se reducía notablemente, mientras que en otros casos empeoraba. Esto confirma que la inclinación de la función de umbral está ciertamente relacionada con la estabilidad en un MCD Sigmoideo, y que utilizar una función para cada neurona puede resultar conveniente. Sin embargo, la variación aleatoria de cada factor no es un enfoque adecuado, pues la estabilidad puede empeorar.

En (Nápoles 2014) se propone un nuevo método de aprendizaje llamado Estabilidad basada en Funciones Sigmoideas (EFS) para mejorar la convergencia en MCD. El objetivo es encontrar la inclinación adecuada λ_i para cada función f_i (con $h_i = 0.5$) de forma tal que se minimice la variabilidad en las salidas del mapa para cada estímulo inicial. Esto conduce a un complejo problema de optimización continuo con tantas variables como conceptos involucre el sistema. Este trabajo apostó por un enfoque heurístico, concretamente Optimización de Enjambres de Partículas (PSO, por sus siglas en inglés), para el cual se propuso un operador para atenuar los efectos negativos que inducen los estados de no progreso en las metaheurísticas poblacionales. El método de aprendizaje EFS y los resultados de las simulaciones para problemas de predicción y clasificación de patrones obtenidos en (Nápoles 2014) constituyen los antecedentes del algoritmo que se presenta en este capítulo.

2.2 DESCRIPCIÓN DE LA PROPUESTA

Esta sección propone un nuevo algoritmo de aprendizaje para mejorar la convergencia en MCD sin afectar la precisión. El objetivo es encontrar la inclinación λ_i y el desplazamiento h_i adecuados para cada función f_i (2.1) de forma tal que se minimice la suma ponderada de la variabilidad de los valores de activación de las neuronas del mapa a través del tiempo para cada estímulo inicial

respecto a los valores esperados y la variabilidad de las salidas reales del mapa respecto a las esperadas. Esto conduce a un complejo problema de optimización continuo cuya cantidad de variables es el doble de los conceptos que involucre el sistema. En principio, este tipo de problemas se puede solucionar con varias técnicas de optimización, sin embargo, este trabajo apuesta por un enfoque heurístico, concretamente metaheurísticas poblacionales.

La razón detrás de esta decisión está relacionada con la complejidad del problema de optimización en sí, pues para un MCD con docenas de conceptos, un enfoque exacto podría resultar demasiado costoso. Por otra parte, las metaheurísticas poblacionales son capaces de estimar soluciones próximas al óptimo global en un tiempo de ejecución razonable, ignorando las propiedades analíticas de la función objetivo (ej. continuidad, convexidad, diferenciabilidad o información del gradiente). Independientemente de estas razones, el algoritmo que se introduce a continuación es extensible para otras técnicas de optimización. (Nápoles 2014)

Para resolver el problema de optimización se necesita codificar los individuos. Cada uno se representa por un vector con $2 * N$ elementos como se muestra:

$$V = (\lambda_1, \lambda_2, \dots, \lambda_N, h_1, h_2, \dots, h_N) \quad (2.2)$$

Los primeros N elementos codifican la inclinación (λ_i) y los últimos N codifican el desplazamiento (h_i) para las respectivas funciones de normalización f_i (ver ecuación 2.1). Ahora, cada nodo C_i se activará usando la i -ésima función de normalización, en lugar de una función fija para todas las neuronas del modelo. Este enfoque, adoptado también en (Nápoles 2014), queda formalizado mediante la ecuación siguiente, permitiendo el proceso de inferencia en el MCD.

$$A_i^{(t+1)} = f_i \left(\sum_{j=1}^N w_{ji} A_j^{(t)} \right) \quad (2.3)$$

Es necesario definir la semántica del método de optimización, que tiene como objetivo reducir al mínimo la suma ponderada de la variabilidad de las respuestas sucesivas del mapa respecto a los valores esperados y la variabilidad de las salidas del mapa respecto a los valores esperados para un conjunto de estímulos iniciales. Para una formulación matemática rigurosa del método se propone la siguiente definición:

Definición 1. Denotamos por $\mathcal{M} = (W, F, A, \tilde{A})$ a un Mapa Cognitivo Difuso donde W es la matriz de pesos causales, $F = (f_1, f_2, \dots, f_N)$ es el conjunto de funciones de transformación asociadas a las neuronas del mapa (siendo N el total de neuronas), A es un tensor de dimensiones $K \times N \times T$ (K es la cantidad de estímulos iniciales y T es el total de iteraciones del mecanismo de inferencia) que contiene, desde el inicio del proceso de inferencia, los valores de activación sucesivos y \tilde{A} es un tensor de dimensiones $K \times N$ que contiene los valores de salida esperados para cada entrada.

Con apoyo de la Definición 1, la ecuación (2.4) muestra la función que debe ser minimizada.

$$\psi(\mathcal{M}) = \alpha * \sum_{k=1}^K \sum_{i=1}^N \sum_{t=1}^T \frac{2\vartheta_t (A_{ki}^{(t)} - \tilde{A}_{ki})^2}{NK(T+1)} + (1 - \alpha) * \sum_{k=1}^K \sum_{i=1}^N \frac{(A_{ki}^{(T)} - \tilde{A}_{ki})^2}{NK} \quad (2.4)$$

Donde:

α : valor constante definido en (0,1) que permite favorecer convergencia o precisión en la función a optimizar en el proceso de búsqueda

ϑ_t : $(t + 1)/T$ Permite restar importancia a las primeras iteraciones del mapa

$A_{ki}^{(t)}$: es el valor de activación de la i -ésima neurona para la k -ésima entrada (estímulo inicial) en la t -ésima iteración

\tilde{A}_{ki} : es la salida esperada de la i -ésima neurona para la k -ésima entrada (estímulo inicial)

Con esta función se pretende superar la principal insuficiencia de los métodos mencionados en el Capítulo 1, que está relacionada con la incapacidad de predecir múltiples instancias del problema modelado (Nápoles, Bello, et al. 2014). El mínimo global de esta función puede corresponderse con varios estados del mapa en cuestión. Los posibles escenarios son:

1. El mapa converge hacia un estado atractor de equilibrio:
 - i. Único para todas las entradas
 - ii. No es único para todas las entradas (opción deseada)
2. El mapa es inestable.
 - i. Es cíclico
 - ii. Es caótico

Analizando los posibles escenarios se observa que el hecho de encontrar el mínimo global de esta función no asegura que el mapa converja y que tenga buena precisión. Puede ocurrir que, al favorecer la convergencia en el proceso de búsqueda, las mejores soluciones encontradas hagan que el mapa establezca hacia el mismo estado atractor de equilibrio. También puede suceder que, favoreciendo la precisión, se encuentren configuraciones inestables, ya que puede ocurrir que en la iteración T (última iteración del proceso de inferencia) el mapa sea muy preciso, aunque no converja. A causa de los posibles escenarios anteriormente descritos, se propone una función de penalización para $\psi(\mathcal{M})$, que permita encontrar las configuraciones deseadas sin caer en estados de inestabilidad o de atractores de equilibrio únicos.

$$P(\mathcal{M}, \psi) = \begin{cases} 1 & \text{si } \mathcal{M} \text{ es inestable} \\ 1 & \text{si } \mathcal{M} \text{ converge a un atractor de equilibrio } \text{único} \\ \psi(\mathcal{M}) & \text{en otro caso} \end{cases} \quad (2.5)$$

Aunque en (2.5) se pasa como argumento a $P(\dots)$ el mapa \mathcal{M} , en el proceso de optimización solo se optimiza el componente F , ya que las relaciones causales entre los nodos (W) se suponen estimadas (a través de expertos, usando algoritmos de aprendizaje o combinando ambas estrategias) y deben ser respetadas para mantener la interpretabilidad del modelo.

¿Por qué utilizar en esta propuesta solo funciones sigmoideas? En principio es posible formular un algoritmo más completo que incluya otras funciones de normalización (ej. binaria, trivalente o basadas en tangentes hiperbólicas). De esta forma no solo se optimizarían la inclinación y desplazamiento adecuados de la función sigmoidea, sino que también sería necesario determinar el tipo de función asociada a cada concepto del mapa. (Nápoles 2014)

Como las funciones discretas solo expresan una cantidad finita de estados para cada concepto (ej. 2, 3), esta variante no es adecuada para todos los casos. Pero una función sigmoidea puede ser discretizada estableciendo un umbral de corte adecuado al problema. En (Bueno & Salmeron 2009) se concluye que las funciones sigmoideas tienen una capacidad de inferencia superior respecto a las demás usadas en la literatura.

2.3 SELECCIÓN DEL OPTIMIZADOR

En principio es posible usar cualquier metaheurística continua, ya que el objetivo principal del algoritmo propuesto es lograr la estabilidad y precisión del mapa, no la selección del mejor método de búsqueda heurístico para optimizar la función descrita en (2.5). Este trabajo apuesta, al igual que en (Nápoles 2014), por la Optimización de Enjambres de Partículas debido a su alta tasa de convergencia y su capacidad de solucionar problemas complejos de optimización global con diferentes características (Wang et al. 2011).

2.3.1 OPTIMIZACIÓN DE ENJAMBRES DE PARTÍCULAS

La metaheurística PSO es un procedimiento de búsqueda bioinspirado en el comportamiento de las poblaciones de pájaros. Este método de búsqueda fue introducido por (Kennedy & Eberhart 1995) para solucionar problemas de optimización continuos. Durante la búsqueda, PSO explota un conjunto de agentes (llamados partículas) que se desplazan a través del espacio de soluciones, tratando de localizar regiones prometedoras. Entonces una partícula se denota como un vector en un espacio N -dimensional, o sea, una posible solución del problema.

En el proceso de búsqueda la posición de una partícula X_i se actualiza en cada ciclo usando una combinación entre la atracción que ejerce el mejor registro personal P_i del agente y la atracción que ejerce la mejor solución global G encontrada por toda población en la vecindad (Bratton & Kennedy 2007). Estos atractores son la base del tercer elemento: la velocidad V_i asociada a cada partícula del enjambre, la cual controla la dirección y magnitud del desplazamiento del individuo en el espacio de búsqueda. La siguiente figura ilustra gráficamente como se actualiza la posición de cada partícula X_i en el espacio desde el punto de vista algebraico.

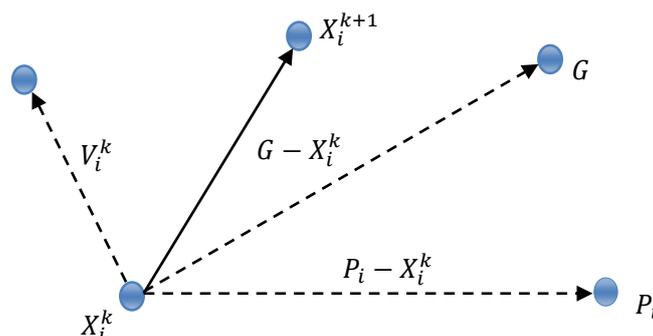


Figura 12. Actualización de la posición de una partícula durante la búsqueda.

Las ecuaciones (2.6) y (2.7) formalizan matemáticamente las reglas para actualizar la posición de los individuos, donde k denota las iteraciones, $\delta_1 \sim U(0, c_1)$ y $\delta_2 \sim U(0, c_2)$ son dos números aleatorios con distribución uniforme. Los factores c_1 y c_2 representan los coeficientes cognitivo y social respectivamente. Estos parámetros deben ser especificados por el usuario. Los coeficientes de aceleración c_1 y c_2 determinan la magnitud de las fuerzas en dirección al registro personal de cada individuo y de la mejor solución encontrada por la bandada. La correcta estimación de estos valores resulta central para garantizar el rendimiento de la metaheurística.

$$V_i^{k+1} = V_i^k + \delta_1(P_i - X_i^k) + \delta_2(G - X_i^k) \quad (2.6)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (2.7)$$

Otro parámetro del método es la velocidad máxima V_M . Este es un importante factor que se usa para acotar la velocidad de las partículas. La selección incorrecta del valor de V_M puede afectar el rendimiento global del algoritmo (Poli et al. 2007). Como una alternativa, en (Shi & Eberhart 1998) los autores presentaron el peso de inercia ω para reemplazar V_M como muestra la siguiente ecuación (2.8). Este factor regula la influencia de la velocidad previa de las partículas durante la actualización de la posición de cada individuo en el espacio de búsqueda.

$$V_i^{k+1} = \omega V_i^k + \delta_1(P_i - X_i^k) + \delta_2(G - X_i^k) \quad (2.8)$$

La introducción de este parámetro es una alternativa para balancear la habilidad de exploración y explotación del algoritmo. Valores altos ($\omega = 0.9$) facilitarán la exploración de nuevas áreas del espacio de soluciones, mientras que valores pequeños ($\omega = 0.1$) son adecuados para explotar las zonas prometedoras. Incluso varios trabajos sugieren estimar el valor de este parámetro de forma dinámica (Nápoles et al. 2012). Por ejemplo, resultarían convenientes valores altos al principio de la búsqueda y valores pequeños al final, garantizando mayor explotación.

A pesar de las ventajas de utilizar este peso de inercia, determinar su valor exacto puede resultar muy complejo en algunos problemas. Incluso algunos autores sugieren combinar esta variante con el método de velocidad acotada, para reducir el efecto de la incorrecta selección del valor de este

parámetro. Como una alternativa, en (Clerc & Kennedy 2002) los autores propusieron dos nuevas variantes conocidas como coeficiente de restricción. La siguiente ecuación (2.9) muestra el coeficiente de restricción Tipo 1. Esta implementación es posiblemente la más difundida entre los investigadores debido a la robustez y simplicidad de su implementación.

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (2.9)$$

En la anterior ecuación $\varphi = c_1 + c_2$. Esto significa que el coeficiente de restricción χ dependerá del valor de los factores de aceleración del algoritmo. La siguiente ecuación (2.10) muestra cómo actualizar las velocidades de las partículas según este modelo. En resumen, si $\varphi > 4$ entonces es posible prevenir la explosión de la nube de partículas (fenómeno donde los agentes y velocidades tienden al infinito). De esta forma se asegura la convergencia del algoritmo (Clerc & Kennedy 2002). A pesar de las garantías del resultado, en (Poli et al. 2007) los autores sugieren acotar las velocidades de las partículas como un enfoque complementario. Un criterio bastante aceptado es fijar la velocidad máxima como la mitad del rango del espacio de soluciones.

$$V_i^{k+1} = \chi \left(V_i^k + \delta_1 (P_i - X_i^k) + \delta_2 (G - X_i^k) \right) \quad (2.10)$$

Pero si $\varphi < 4$ entonces no es posible garantizar la convergencia del algoritmo. Incluso es posible que las partículas muestren comportamientos discretos. Por ejemplo, (Clerc & Kennedy 2002) notaron que si $\varphi = (5 + \sqrt{5})/2$ entonces una partícula no aleatoria en \mathbb{R} se desplazaría de forma cíclica a través de 5 estados, independientemente de la naturaleza de la metaheurística. Si alguno de estos estados no representa un óptimo, entonces esto desencadenará irremediablemente en una configuración de estancamiento, donde no hay progresos en la búsqueda. Otro escenario curioso que viola estas condiciones tiene lugar cuando $\varphi = 5 + \sqrt{3}$. En este caso específico una partícula en un espacio 1-dimensional tendrá un comportamiento cuasi-cíclico.

Normalmente se toma $c_1 = c_2 = 2.05$ lo que garantiza que $\varphi > 4$. Observe que el coeficiente de restricción χ es equivalente al peso de inercia, si $c_1 = c_2 = 1.4961$ y $\omega = 0.7298$. Estos valores serán adoptados en el resto del trabajo, pues en otros estudios han demostrado ser una alternativa robusta para diferentes tipos de problemas (Nápoles, Grau, et al. 2014). Por lo tanto, este

optimizador solo requiere que el usuario determine el número máximo de evaluaciones de la función objetivo permitidas para la ejecución del algoritmo y el tamaño de la población.

Más simplificada, basta garantizar estas condiciones de convergencia para asegurar que la metaheurística tendrá un comportamiento adecuado. Esto reduce elocuentemente el esfuerzo que requiere la configuración de los parámetros sensibles a un modelo. De hecho, el trabajo de (Clerc & Kennedy 2002) da motivo al autor para usar este método como optimizador.

2.3.2 PSO CON TOPOLOGÍAS ESTRUCTURADAS

Otro aspecto relevante en la metaheurística PSO es la elección del atractor global. Este atractor denota la mejor solución encontrada por los individuos en su vecindad. En PSO una vecindad se define, para cada partícula de la bandada, como el subconjunto de agentes que pueden interactuar con dicha partícula (Bratton & Kennedy 2007). Si la vecindad de cada partícula incluye toda la bandada entonces se está en presencia del modelo global, pero si existen algún tipo de restricción en el protocolo de comunicación se está en presencia de un modelo local.

Al principio las topologías locales no tuvieron mucha aceptación pues su rendimiento no lograba superar al modelo global. Sin embargo, en (Kennedy & Mendes 2006) los autores demostraron que el rendimiento de PSO era superior si se usaba una topología estructurada en conjunción con otras mejoras a la metaheurística. La siguiente Figura 13 muestra el protocolo de comunicación entre las partículas para las dos topologías más utilizadas: la Topología Global y la Topología de Anillo. La Topología de Anillo es el modelo local más simple; donde el agente X_i interactúa con las partículas X_{i-1} y X_{i+1} de la población. De forma general este protocolo es bastante aceptado debido a su habilidad para preservar la diversidad entre los individuos.

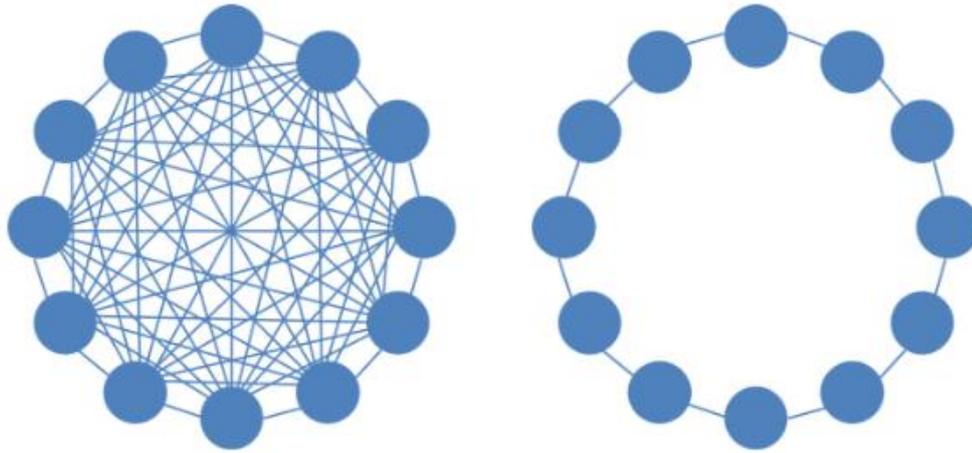


Figura 13. Diferentes protocolos de comunicación entre las partículas de la población. El primer ejemplo muestra la Topología Global (GBest), mientras que el segundo ejemplo ilustra la Topología de Anillo (LBest).

Por otra parte, en (Li 2010) se demostró que un PSO con Topología de Anillo podía converger a múltiples óptimos de forma simultánea, sin nuevos operadores. Durante el proceso de búsqueda la bandada implícitamente se descompone en varias subpoblaciones, como resultado de la escasa interacción entre los agentes. En espacios de búsqueda multimodales, cada subpoblación podría converger a un óptimo diferente (local o global). Eventualmente la información se propagará por toda la bandada, sin embargo, en algunos casos esto no es suficiente.

A pesar de las ventajas evidentes de las variantes locales, en problemas de optimización global es preferible la topología con mayor comunicación. No obstante, la velocidad de convergencia de esta variante frecuentemente conduce a estados de convergencia prematura, como resultado de la pérdida de la diversidad en la población artificial. Por ello se creó la variante híbrida GLBest, que combina las capacidades de convergencia de la topología global con la preservación de diversidad presente en las topologías locales (ej. topología de anillo).

2.4 DEFINICIONES SOBRE CONVERGENCIA Y PRECISIÓN

En la actualidad, según el conocimiento del autor, no ha sido encontrado ningún método simple, rápido y confiable para predecir la estabilidad de un MCD, ya que el análisis de estabilidad de un MCD recae en la simulación de múltiples escenarios (Carvalho & Tomé 2002). Este es un complejo problema matemático sobre el cual el propio Kosko afirma que permanecerá insoluble en los próximos años. Los algoritmos de aprendizaje para MCD están dirigidos a estimar una matriz de pesos que maximice el rendimiento del sistema, y entre estos existen varios cuyo objetivo

es minimizar el error de precisión (ej. basados en metaheurísticas poblacionales). Mientras los algoritmos para la mejora de la estabilidad de MCD (propuestos como antecedente y contribución de este trabajo) parten de una matriz de pesos ya aprendida, todos estos métodos son costosos computacionalmente y adicionalmente requieren un conjunto de pares entrada-salida (a cada estímulo de entrada se le hace corresponder la salida esperada). Con las definiciones propuestas a continuación se pretende, no solo comprobar la convergencia del MCD, sino también la precisión, de una forma sencilla y con menor costo computacional que el necesario en la simulación de varios escenarios.

2.4.1 MAPAS COGNITIVOS DIFUSOS ε -ESTABLES Y ε -INESTABLES

Usando la Definición 1 como base, se proponen varios fundamentos que dan paso a las formulaciones sobre ε -estabilidad y ε -inestabilidad de Mapas Cognitivos Difusos.

Definición 2. Se dice que una neurona A_{ki} es ε -precisa en la t -ésima iteración del mapa \mathcal{M} si se cumple que $|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon \quad \forall k$ (con $\varepsilon \in \mathbf{R} : \varepsilon \geq 0$).

Definición 3. Se dice que el mapa \mathcal{M} es ε -preciso en la t -ésima iteración si toda neurona A_{ki} es ε -precisa en la t -ésima iteración.

Definición 4. Se dice que $\mathcal{M} = (W, F, A, \tilde{A})$ es ε -estable si:

$$\exists t_q \in \mathbb{N} : |A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon \quad \forall i, \forall k, \forall t \geq t_q \quad \text{donde } \varepsilon \in \mathbf{R} : \varepsilon \geq 0$$

El mapa \mathcal{M} es ε -estable si a partir de cierta iteración t_q (las iteraciones son valores naturales), es ε -preciso en la t -ésima iteración $\forall t \geq t_q$. Tomando como ejemplo problemas de simulación de escenarios, son deseables mapas ε -estables con ε tan cerca de 0 como sea posible. Se puede asegurar que un mapa tiene mejor estabilidad y precisión cuando ε se acerca a 0. Sin correr el mecanismo de inferencia para los escenarios deseados, es muy difícil encontrar el menor ε tal que el mapa sea ε -estable. Incluso haciendo esto, el ε óptimo puede ser imposible de encontrar. Este valor puede ser hallado cuando el mapa en cuestión es estable o cíclico, pero en el caso caótico no

es suficiente con iterar el mapa, ya que en las infinitas iteraciones posibles del mecanismo de inferencia puede aparecer siempre un valor menor para ε .

Corolario 4.1. Si $\mathcal{M} = (W, F, A, \tilde{A})$ es ε -estable entonces también es α -estable $\forall \alpha \in \mathbf{R} : \alpha > \varepsilon$.

Corolario 4.2. Todo $\mathcal{M} = (W, F, A, \tilde{A})$ es 1-estable.

Definición 5. Se dice que $\mathcal{M} = (W, F, A, \tilde{A})$ es ε -inestable si:

$$\nexists t_q \in \mathbb{N} : |A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon \quad \forall i, \forall k, \forall t \geq t_q \quad \text{donde } \varepsilon \in \mathbf{R} : \varepsilon \geq 0$$

Un mapa es ε -inestable si no es ε -estable, y viceversa. La ε -inestabilidad implica que al menos una neurona del mapa no es ε -precisa en la t -ésima iteración del mecanismo de inferencia.

Corolario 5.1. Si $\mathcal{M} = (W, F, A, \tilde{A})$ es ε -inestable entonces también es α -inestable $\forall \alpha \in \mathbf{R} : \alpha < \varepsilon$.

Usando la Definición 2 se pueden generalizar las definiciones 3, 4 y 5, de forma tal que para cada neurona se establezca una ε -precisión diferente. De esta forma un mapa puede ser E -preciso, donde E es el conjunto de las precisiones asociadas a cada una de las neuronas. Siendo $E = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N\}$, se pueden plantear las generalizaciones antes mencionadas.

Definición 6. Se dice que el mapa \mathcal{M} es E -preciso en la t -ésima iteración si toda neurona A_{ki} es ε_i -precisa en la t -ésima iteración.

Definición 7. Se dice que $\mathcal{M} = (W, F, A, \tilde{A})$ es E -estable si:

$$\exists t_q \in \mathbb{N} : |A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \quad \forall i, \forall k, \forall t \geq t_q \quad \text{donde } \varepsilon_i \in \mathbf{R} : \varepsilon_i \geq 0$$

El mapa \mathcal{M} es E -estable si a partir de cierta iteración t_q (las iteraciones son valores naturales), es E -preciso en la t -ésima iteración $\forall t \geq t_q$. Tomando como ejemplo problemas de simulación de escenarios, son deseables mapas E -estables con $Prom(E) = \frac{\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_N}{N}$ tan cerca de 0 como sea posible. Se puede asegurar que un mapa tiene mejor estabilidad y precisión cuando $Prom(E)$ se acerca a 0. Sin correr el mecanismo de inferencia para los escenarios deseados, es muy difícil encontrar el menor $Prom(E)$ tal que el mapa sea E -estable. Este problema comparte las mismas dificultades que el de demostrar la ε -estabilidad, siendo ligeramente superiores, ya que se debe encontrar no solo un valor, sino un conjunto cuya cardinalidad es igual al tamaño (cantidad de neuronas) del mapa en cuestión.

Corolario 7.1. Si $\mathcal{M} = (W, F, A, \tilde{A})$ es E -estable entonces también es J -estable, siendo $J = \{j_1, j_2, \dots, j_N\}$ y $j_i \geq \varepsilon_i \forall i$.

Definición 8. Se dice que $\mathcal{M} = (W, F, A, \tilde{A})$ es E -inestable si:

$$\nexists t_q \in \mathbb{N} : |A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \quad \forall i, \forall k, \forall t \geq t_q \quad \text{donde } \varepsilon_i \in \mathbf{R} : \varepsilon_i \geq 0$$

Corolario 8.1. Si $\mathcal{M} = (W, F, A, \tilde{A})$ es E -inestable entonces también es J -inestable, siendo $J = \{j_1, j_2, \dots, j_N\}$ y $j_i \leq \varepsilon_i \forall i$.

Dada la complejidad de la comprobación de estas definiciones, se proponen condiciones de suficiencia para el cumplimiento de ambas (Definición 7 y Definición 8), en las cuales no hay necesidad de iterar el mapa. En las condiciones de suficiencia para la Definición 7 es necesario el conocimiento de los vectores de activación iniciales y las salidas esperadas para cada uno de los anteriores (cada vector tiene los valores asociados a todas las neuronas del mapa). Se puede iterar el mapa para poder comprobar que este sea E -preciso en alguna iteración, aunque se pueden tomar los valores de activación iniciales con este fin.

Condición de Suficiencia 1 para Definición 7. Dado $\mathcal{M} = (W, F, A, \tilde{A})$, las condiciones siguientes deben cumplirse simultáneamente para ser E -estable:

$$(|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \quad \forall i, \forall k) \xrightarrow{\text{implica}} (|MIN_i - \tilde{A}_{ki}| \leq \varepsilon_i) \text{ y } (|MAX_i - \tilde{A}_{ki}| \leq \varepsilon_i) \quad \forall i, \forall k \quad (2.11)$$

$$\exists t_q : \mathcal{M} \text{ es } E\text{-preciso en la } t_q\text{-ésima iteración} \quad (2.12)$$

Donde:

$$MIN_i = \frac{1}{1 + e^{-\lambda_i(w_i \tilde{A}_k + \min(w_i D_k^t) - h_i)}} \quad (2.13)$$

$$MAX_i = \frac{1}{1 + e^{-\lambda_i(w_i \tilde{A}_k + \max(w_i D_k^t) - h_i)}} \quad (2.14)$$

$$w_i = [w_{i1}, w_{i2}, \dots, w_{in}] \quad (2.15)$$

$$\tilde{A}_k = [\tilde{A}_{k1}, \tilde{A}_{k2}, \dots, \tilde{A}_{kn}] \quad (2.16)$$

$$D_k^t = [d_{k1}^t, d_{k2}^t, \dots, d_{kn}^t] \quad (2.17)$$

Demostración. La condición (2.11) quiere decir que el hecho de que \mathcal{M} sea E -preciso en la t -ésima iteración, implica que lo sea también en la $(t + 1)$ -ésima iteración. Inductivamente, si se cumple para t , entonces se cumple para todas las iteraciones siguientes. Como inicio de inducción se tiene la condición (2.12), que asegura que \mathcal{M} es E -preciso en la t_q -ésima iteración, cumpliéndose así la Definición 7. Para demostrar esto se comienza analizando que el mecanismo de inferencia de \mathcal{M} actualiza los valores de activación de cada neurona como se muestra:

$$A_{ki}^{t+1} = \frac{1}{1 + e^{-\lambda_i(w_i A_k^t - h_i)}} \quad \forall i, \forall k \quad (2.18)$$

Donde:

$$A_k^t = [A_{k1}^t, A_{k2}^t, \dots, A_{kn}^t] \quad (2.19)$$

Sabiendo que $|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \quad \forall i, \forall k$ para algún t , se puede decir que:

$$A_{ki}^t = \tilde{A}_{ki} + d_{ki}^t \quad \text{con} \quad -\varepsilon_i \leq d_{ki}^t \leq \varepsilon_i \quad \forall i, \forall k \quad (2.20)$$

Se reescribe (2.18) como se muestra:

$$A_{ki}^{t+1} = \frac{1}{1 + e^{-\lambda_i(w_i \cdot \tilde{A}_k + w_i \cdot D_k^t - h_i)}} \quad \forall i, \forall k \quad (2.21)$$

Dado que $f_i(x) = \frac{1}{1 + e^{-\lambda_i(x-h_i)}}$ es una función creciente, se establece que:

$$\frac{1}{1 + e^{-\lambda_i(w_i \cdot \tilde{A}_k + \min(w_i \cdot D_k^t) - h_i)}} \leq A_{ki}^{t+1} \leq \frac{1}{1 + e^{-\lambda_i(w_i \cdot \tilde{A}_k + \max(w_i \cdot D_k^t) - h_i)}} \quad \forall i, \forall k \quad (2.22)$$

Se denota por $sgn(\cdot)$ a la función que retorna -1 si el argumento es negativo, 1 si es positivo y 0 si el argumento es 0. Usando la condición (2.20) resulta obvio que:

$$\min(w_i \cdot D_k^t) = \sum_{j=1}^n -sgn(w_{ij}) w_{ij} \varepsilon_i \quad (2.23)$$

$$\max(w_i \cdot D_k^t) = \sum_{j=1}^n sgn(w_{ij}) w_{ij} \varepsilon_i \quad (2.24)$$

De esta forma, se tiene un intervalo para $A_{ki}^{t+1} \quad \forall i, \forall k$. Y si se cumple que:

$$(|MIN_i - \tilde{A}_{ki}| \leq \varepsilon_i) \quad y \quad (|MAX_i - \tilde{A}_{ki}| \leq \varepsilon_i) \quad \forall i, \forall k \quad (2.25)$$

Entonces:

$$\mathcal{M} \text{ es } E\text{-preciso en la } (t+1)\text{-ésima iteración} \quad (2.26)$$

Usando la condición (2.12) se tiene que $\exists t_q : \mathcal{M}$ es E -preciso en la t -ésima iteración y entonces \mathcal{M} es E -preciso en las iteraciones siguientes (2.26), por lo que \mathcal{M} es E -estable. Finalmente, el mapa \mathcal{M} es E -estable si se verifican:

$$(|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \quad \forall i, \forall k) \xrightarrow{\text{implica}} (|MIN_i - \tilde{A}_{ki}| \leq \varepsilon_i) \quad y \quad (|MAX_i - \tilde{A}_{ki}| \leq \varepsilon_i) \quad \forall i, \forall k$$

Y:

$$\exists t_q : \mathcal{M} \text{ es } E\text{-preciso en la } t_q\text{-ésima iteración}$$

■

Condición de Suficiencia 2 para Definición 7. Dado $\mathcal{M} = (W, F, A, \tilde{A})$, tal que el intervalo para todos los valores de inclinación es $[\lambda_{min}, \lambda_{max}]$ y para el desplazamiento es $[h_{min}, h_{max}]$, las condiciones siguientes deben cumplirse simultáneamente para ser E -estable:

$$(|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \quad \forall i, \forall k) \xrightarrow{\text{implica}} (|MIN_i - \tilde{A}_{ki}| \leq \varepsilon_i) \text{ y } (|MAX_i - \tilde{A}_{ki}| \leq \varepsilon_i) \quad \forall i, \forall k \quad (2.27)$$

$$\exists t_q : \mathcal{M} \text{ es } E\text{-preciso en la } t_q\text{-ésima iteración} \quad (2.28)$$

Donde:

$$MIN_i = \begin{cases} \frac{1}{1 + e^{-\lambda_{max}(w_i \cdot \tilde{A}_k + \min(w_i \cdot D_k^t) - h_{max})}}, & \text{si } w_i \cdot \tilde{A}_k + \min(w_i \cdot D_k^t) - h_{max} \leq 0 \\ \frac{1}{1 + e^{-\lambda_{min}(w_i \cdot \tilde{A}_k + \min(w_i \cdot D_k^t) - h_{max})}}, & \text{si } w_i \cdot \tilde{A}_k + \min(w_i \cdot D_k^t) - h_{max} > 0 \end{cases} \quad (2.29)$$

$$MAX_i = \begin{cases} \frac{1}{1 + e^{-\lambda_{min}(w_i \cdot \tilde{A}_k + \max(w_i \cdot D_k^t) - h_{min})}}, & \text{si } w_i \cdot \tilde{A}_k + \max(w_i \cdot D_k^t) - h_{min} \leq 0 \\ \frac{1}{1 + e^{-\lambda_{max}(w_i \cdot \tilde{A}_k + \max(w_i \cdot D_k^t) - h_{min})}}, & \text{si } w_i \cdot \tilde{A}_k + \max(w_i \cdot D_k^t) - h_{min} > 0 \end{cases} \quad (2.30)$$

Demostración. Se usan los mismos términos que en la demostración anterior, a excepción de MIN_i y MAX_i , que se redefinieron previamente.

Como MIN_i y MAX_i son los valores extremos para A_{ki}^{t+1} se tiene que:

$$MIN_i \leq A_{ki}^{t+1} \leq MAX_i \quad \forall i, \forall k \quad (2.31)$$

De esta forma, se tiene un intervalo para $A_{ki}^{t+1} \quad \forall i, \forall k$. Y si se cumple que:

$$(|MIN_i - \tilde{A}_{ki}| \leq \varepsilon_i) \text{ y } (|MAX_i - \tilde{A}_{ki}| \leq \varepsilon_i) \quad \forall i, \forall k \quad (2.32)$$

Entonces:

$$\mathcal{M} \text{ es } E\text{-preciso en la } (t+1)\text{-ésima iteración} \quad (2.33)$$

Usando la condición (2.28) se tiene que $\exists t_q : \mathcal{M}$ es E -preciso en la t -ésima iteración y entonces \mathcal{M} es E -preciso en las iteraciones siguientes (2.33), por lo que \mathcal{M} es E -estable. Finalmente, el mapa \mathcal{M} es E -estable si se verifican:

$$\left(|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \quad \forall i, \forall k \right) \xrightarrow{\text{implica}} \left(|MIN_i - \tilde{A}_{ki}| \leq \varepsilon_i \right) \text{ y } \left(|MAX_i - \tilde{A}_{ki}| \leq \varepsilon_i \right) \quad \forall i, \forall k$$

Y:

$$\exists t_q : \mathcal{M} \text{ es } E\text{-preciso en la } t_q\text{-ésima iteración}$$

■

En las condiciones de suficiencia para la Definición 8 no es necesario el conocimiento de los vectores de activación iniciales, sino las salidas esperadas para cada uno de los anteriores (cada vector tiene los valores asociados a todas las neuronas del mapa).

Condición de Suficiencia 1 para Definición 8. Dado $\mathcal{M} = (W, F, A, \tilde{A})$, la condición siguiente debe cumplirse para ser E -inestable:

$$\left(|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \quad \forall i, \forall k \right) \xrightarrow{\text{implica}} \exists j, k : \left(\tilde{A}_{kj} < MIN_j - \varepsilon_j \right) \text{ o } \left(\tilde{A}_{kj} > MAX_j + \varepsilon_j \right) \quad (2.34)$$

Donde:

$$MIN_j = \frac{1}{1 + e^{-\lambda_j(w_j \cdot \tilde{A}_k + \min(w_j \cdot D_k^t) - h_j)}} \quad (2.35)$$

$$MAX_j = \frac{1}{1 + e^{-\lambda_j(w_j \cdot \tilde{A}_k + \max(w_j \cdot D_k^t) - h_j)}} \quad (2.36)$$

Demostración. Se usan los mismos términos que en las demostraciones anteriores, a excepción de MIN_j y MAX_j , que se redefinieron previamente.

Como MIN_j y MAX_j , son los valores extremos para A_{kj}^{t+1} se tiene que:

$$MIN_j \leq A_{kj}^{t+1} \leq MAX_j \quad \forall j, \forall k \quad (2.37)$$

De esta forma, se tiene un intervalo para $A_{kj}^{t+1} \quad \forall j, \forall k$. Y si se cumple que:

$$\exists j, k : (\tilde{A}_{kj} < MIN_j - \varepsilon_j) \text{ o } (\tilde{A}_{kj} > MAX_j + \varepsilon_j) \quad (2.38)$$

Entonces:

$$\mathcal{M} \text{ no es } E\text{-preciso en la } (t+1)\text{-ésima iteración} \quad (2.39)$$

Por el resultado (2.39) se tiene que es imposible que \mathcal{M} sea E -estable. Finalmente, el mapa \mathcal{M} es E -inestable si se verifica:

$$(|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \forall i, \forall k) \xrightarrow{\text{implica}} \exists j, k : (\tilde{A}_{kj} < MIN_j - \varepsilon_j) \text{ o } (\tilde{A}_{kj} > MAX_j + \varepsilon_j)$$

■

Condición de Suficiencia 2 para Definición 8. Dado $\mathcal{M} = (W, F, A, \tilde{A})$, tal que el intervalo para todos los valores de inclinación es $[\lambda_{min}, \lambda_{max}]$ y para el desplazamiento es $[h_{min}, h_{max}]$, la condición siguiente debe cumplirse para ser E -inestable:

$$(|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \forall i, \forall k) \xrightarrow{\text{implica}} \exists j, k : (\tilde{A}_{kj} < MIN_j - \varepsilon_j) \text{ o } (\tilde{A}_{kj} > MAX_j + \varepsilon_j) \quad (2.40)$$

Donde:

$$MIN_j = \begin{cases} \frac{1}{1 + e^{-\lambda_{max}(w_j \cdot \tilde{A}_k + \min(w_j \cdot D_k^t) - h_{max})}}, & \text{si } w_j \cdot \tilde{A}_k + \min(w_j \cdot D_k^t) - h_{max} \leq 0 \\ \frac{1}{1 + e^{-\lambda_{min}(w_j \cdot \tilde{A}_k + \min(w_j \cdot D_k^t) - h_{max})}}, & \text{si } w_j \cdot \tilde{A}_k + \min(w_j \cdot D_k^t) - h_{max} > 0 \end{cases} \quad (2.41)$$

$$MAX_j = \begin{cases} \frac{1}{1 + e^{-\lambda_{min}(w_j \cdot \tilde{A}_k + \max(w_j \cdot D_k^t) - h_{min})}}, & \text{si } w_j \cdot \tilde{A}_k + \max(w_j \cdot D_k^t) - h_{min} \leq 0 \\ \frac{1}{1 + e^{-\lambda_{max}(w_j \cdot \tilde{A}_k + \max(w_j \cdot D_k^t) - h_{min})}}, & \text{si } w_j \cdot \tilde{A}_k + \max(w_j \cdot D_k^t) - h_{min} > 0 \end{cases} \quad (2.42)$$

Demostración. Se usan los mismos términos que en las demostraciones anteriores, a excepción de MIN_j y MAX_j , que se redefinieron previamente.

Como MIN_j y MAX_j , son los valores extremos para A_{kj}^{t+1} se tiene que:

$$MIN_j \leq A_{kj}^{t+1} \leq MAX_j \quad \forall j, \forall k \quad (2.43)$$

De esta forma, se tiene un intervalo para $A_{kj}^{t+1} \forall j, \forall k$. Y si se cumple que:

$$\exists j, k : (\tilde{A}_{kj} < MIN_j - \varepsilon_j) \text{ o } (\tilde{A}_{kj} > MAX_j + \varepsilon_j) \quad (2.44)$$

Entonces:

$$\mathcal{M} \text{ no es } E\text{-preciso en la } (t + 1)\text{-ésima iteración} \quad (2.45)$$

Por el resultado (2.45) se tiene que es imposible que \mathcal{M} sea E -estable. Finalmente, el mapa \mathcal{M} es E -inestable si se verifica:

$$(|A_{ki}^t - \tilde{A}_{ki}| \leq \varepsilon_i \forall i, \forall k) \xrightarrow{\text{implica}} \exists j, k : (\tilde{A}_{kj} < MIN_j - \varepsilon_j) \text{ o } (\tilde{A}_{kj} > MAX_j + \varepsilon_j)$$

■

2.5 CONCLUSIONES PARCIALES

Sin la estabilidad en los MCD no es posible la toma de decisiones acertada, lo que afecta la usabilidad de estas estructuras neuronales. A partir de los sistemas modelados o de los algoritmos de aprendizaje existentes, no hay seguridad de obtener mapas estables. Aunque los métodos analíticos enunciados en el Capítulo 1 permiten determinar condiciones para asegurar convergencia en MCD Sigmoideos, estos no consideran la precisión.

Se propuso un algoritmo que permite mejorar la convergencia y la precisión en MCD, luego de que la matriz de pesos se ha estimado. Este método no garantiza la convergencia del sistema a un estado atractor de equilibrio para todos los estímulos iniciales, pero es una alternativa factible ante los problemas de inestabilidad en MCD, sin afectar la precisión.

Para solucionar el problema de optimización asociado, se seleccionó el algoritmo GLBest, que es un híbrido entre el PSO con topología global (GBest) y el PSO con topología de anillo (LBest). En el próximo capítulo se fundamentará a través de la evaluación experimental la decisión de escoger este método sobre las variantes de PSO: GBest y LBest.

Se propusieron varias definiciones que dan un punto de vista diferente para analizar la estabilidad y precisión en los MCD Sigmoideos. También se plantean condiciones de suficiencia que permiten reducir la complejidad computacional asociada a la comprobación directa de las definiciones antes mencionadas. Estos resultados constituyen un acercamiento analítico al problema de la convergencia en MCD Sigmoideos, a la vez que se trata el problema de la precisión. Además, en el próximo capítulo, serán usados para evaluar los resultados obtenidos por el algoritmo de optimización escogido.

CAPÍTULO 3

EVALUACIÓN DEL ALGORITMO
DE APRENDIZAJE

CAPÍTULO 3. EVALUACIÓN DEL ALGORITMO DE APRENDIZAJE

En este capítulo se presentan experimentos y simulaciones que permiten evaluar el rendimiento del algoritmo de optimización propuesto en el capítulo anterior. Para ello se construye un conjunto de mapas artificiales con diversas características, sobre los cuales se realizarán todos los experimentos. Primero se selecciona la mejor variante de PSO (GBest, LBest, GLBest) para resolver el problema de optimización y luego, usando la variante escogida, se selecciona el α óptimo para ser usado en la función objetivo (2.4). Luego, para validar los resultados obtenidos con la combinación óptima de variante de PSO y α , se seleccionan aleatoriamente 50 mapas para aplicar los resultados analíticos demostrados en el capítulo anterior.

3.1 GENERACIÓN DEL CONJUNTO DE MAPAS DE PRUEBA

Para realizar todos los experimentos se generó un conjunto de 1000 mapas artificiales con varias características, permitiendo la diversidad y cumpliendo con varias restricciones que serán descritas a continuación.

Restricciones para el conjunto de mapas:

1. Se dividen en 10 grupos de acuerdo al porcentaje de conectividad² (10%, 20%, ..., 100%).
2. El número de neuronas de cada mapa es un número aleatorio distribuido uniformemente entre 1 y 100.
3. Las relaciones causales de la matriz de pesos son valores reales distribuidos uniformemente entre -1 y 1.
4. En ningún mapa existen auto-conexiones (una neurona no influye sobre sí misma).
5. Para cada mapa se generan 20 juegos de estímulos iniciales (cada juego consiste en un vector cuyos valores están distribuidos uniformemente entre 0 y 1). Los valores de salida asociados a cada vector se calculan realizando 100 iteraciones del MCD usando la misma función sigmoidea para toda neurona, con parámetros $\lambda_i = 5$ y $h_i = 0$.

² Razón entre el número de conexiones generadas y el total de relaciones causales posibles para el mapa.

- Además, luego de cumplir todos los requisitos anteriores, un mapa es añadido al conjunto solo si es inestable para cada estímulo inicial (20).

3.2 VALIDACIÓN EXPERIMENTAL

La correcta elección del algoritmo de optimización depende en gran medida del espacio de búsqueda descrito por la función objetivo (2.4). Además, permite la justificación de ciertas decisiones como los intervalos de búsqueda para λ_i y h_i , que deben especificarse en el proceso de optimización.

Se toma el grupo de mapas con un 80% de conectividad para ilustrar el comportamiento de la región de búsqueda, observándose resultados similares para otros grupos. Se fijan intervalos para λ_i y h_i , tomando $[0,10]$ y $[0,1]$ respectivamente. De cada intervalo se toman 50 valores igualmente espaciados y, para cada combinación, se realiza el proceso de inferencia para cada mapa (del total de 100), con el fin de obtener el valor de la función objetivo (tomando $T = 100$, $\alpha = 0.5$ y valores fijos para la inclinación y el desplazamiento). Así, para cada combinación (2500) se tienen 100 valores que deben promediarse para generar la región de búsqueda, que se muestra en la Figura 14.

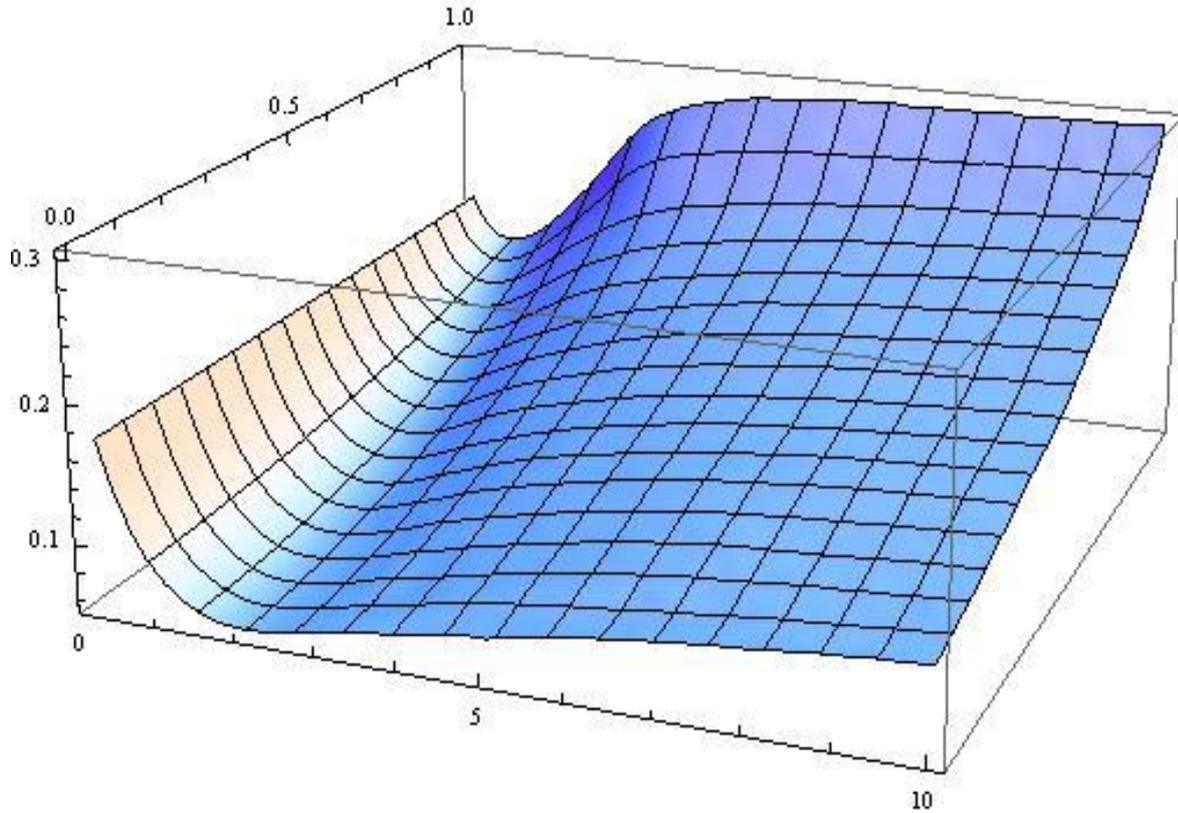


Figura 14. El eje X (a lo profundo) contiene los valores para el desplazamiento (entre 0 y 1), el eje Y (a lo ancho) contiene los valores para la inclinación (entre 0 y 10) y el eje Z (a lo alto) muestra los valores de la función objetivo.

La gráfica demuestra que la elección correcta de estos parámetros tiene implicaciones directas en la convergencia y la precisión de los MCD Sigmoideos, ya que se evidencian zonas con valores muy diversos para (2.4). Por ejemplo, se puede observar como al aumentar λ_i y h_i el valor de (2.4) aumenta y también que con λ_i entre 1 y 3 aproximadamente (sin importar h_i) se alcanzan los valores mínimos.

Es interesante saber para cuáles combinaciones todos los mapas convergen al mismo estado atractor de equilibrio, ya que, sin importar los pesos causales de los mapas, estos siempre convergerán al mismo punto. Estas combinaciones no son deseadas en problemas de simulación de múltiples escenarios. La Figura 15 muestra los puntos sobre la región de búsqueda, para los cuales ocurre la situación descrita anteriormente.

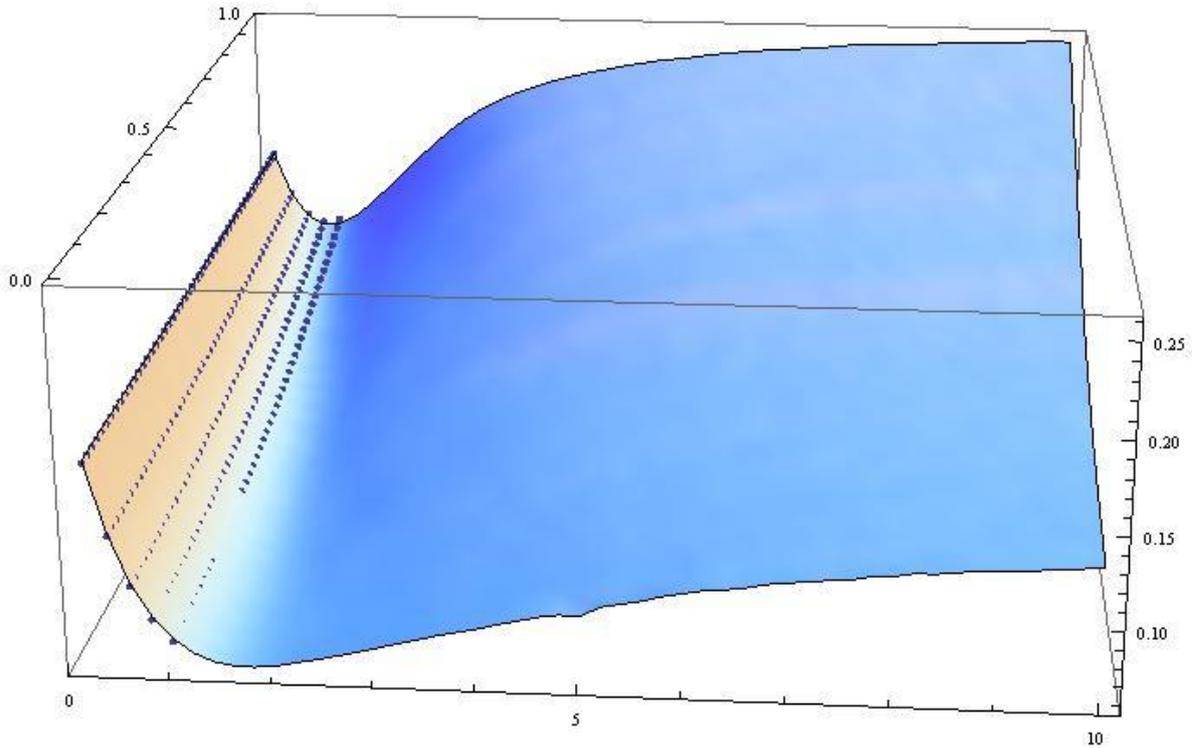


Figura 15. Los puntos fijos únicos sobre el espacio de búsqueda (para cada combinación de parámetros).

Resulta que, sin importar h_i y para $\lambda_i \leq 1$ aproximadamente, casi la totalidad de las combinaciones hacen que todo mapa converja a un único atractor de equilibrio, lo cual no es deseable en problemas de simulación de múltiples escenarios. Este resultado tiene estrecha relación con el Teorema 4 de la sección 1.5, propuesto por (Knight, Lloyd, Penn, et al. 2014).

3.2.1 SELECCIÓN DE TOPOLOGÍA DEL PSO

Para evaluar el rendimiento global del algoritmo GLBest se propone un estudio comparativo que involucra múltiples variantes de PSO. Esta comparación incluye al PSO con topología global (GBest) y al PSO con topología de anillo (LBest).

Para evaluar el rendimiento de estos métodos se usa la función objetivo (2.4) con parámetro $\alpha = 0.5$ (dando igual importancia a la convergencia y la precisión). Además, la configuración de parámetros del experimento se resume como sigue: una bandada artificial de 80 individuos, 200

evaluaciones de la función objetivo como criterio de parada, coeficientes de aceleración $c_1 = 2.05$ y $c_2 = 2.05$ y coeficiente de restricción $c_r = 0.72984$.

En el experimento se toman 50 mapas aleatorios del conjunto artificial y, para cada variante de PSO, se promedian 5 evaluaciones del algoritmo por mapa. La Tabla 1 muestra los resultados obtenidos.

Tabla 1. Óptimos encontrados por las variantes de PSO.

Mapas\Algoritmos	GBest	LBest	GLBest
M1	0.00809402	0.00664684	0.0049539
M2	0.01553171	0.00774017	0.00702883
M3	0.02113618	0.01442662	0.00928753
M4	0.02228547	0.0264959	0.01867959
M5	0.01063566	0.00957554	0.00661248
M6	0.04801784	0.04316175	0.03856869
M7	0.00682579	0.00561142	0.00527135
M8	0.03629306	0.0359315	0.03420356
M9	0.01951705	0.01818958	0.01602494
M10	0.02213347	0.02231617	0.02057219
M11	0.08701446	0.08636092	0.08308754
M12	0.02562259	0.0245727	0.02335031
M13	0.00953007	0.00891105	0.00860259
M14	0.00613055	0.00608787	0.00564619
M15	0.02698451	0.02625179	0.02429144
M16	0.01500205	0.01486893	0.0130571
M17	0.03836703	0.03763752	0.03305943
M18	0.04438472	0.04426784	0.04338789
M19	0.00859459	0.00779316	0.00463021
M20	0.00773214	0.00572841	0.0034215
M21	0.01301732	0.01317528	0.00433091
M22	0.00765521	0.00768083	0.00368407
M23	0.01175042	0.01183056	0.00876702
M24	0.01515247	0.01376398	0.00971261
M25	0.01123241	0.01254927	0.00955977
M26	0.02746908	0.02399816	0.01927114
M27	0.01041477	0.00813297	0.00322748
M28	0.02133529	0.01263857	0.00454376
M29	0.01278977	0.01438676	0.01106642
M30	0.01722176	0.01490166	0.01131545
M31	0.04451595	0.04194375	0.04077525
M32	0.03822034	0.03518518	0.02948676
M33	0.02248833	0.02409526	0.0200769

M34	0.00999589	0.01088391	0.00859748
M35	0.00928555	0.01043492	0.00692537
M36	0.04627422	0.04650576	0.04362524
M37	0.03769418	0.04168435	0.02960228
M38	0.0322332	0.03137133	0.02938211
M39	0.01168832	0.01236421	0.0098973
M40	0.02221813	0.02190352	0.01818432
M41	0.01537779	0.01535949	0.01020226
M42	0.00175543	0.00246232	0.00178313
M43	0.04545463	0.04061176	0.03543242
M44	0.01194905	0.01416698	0.01127173
M45	0.05900921	0.04787112	0.05151409
M46	0.09906518	0.091738	0.0889481
M47	0.02480675	0.02530453	0.02050965
M48	0.03380258	0.0309991	0.02849508
M49	0.0584155	0.05916697	0.0571153
M50	0.03985562	0.03247809	0.02921824
PROMEDIO	0.02583955	0.02444329	0.02120518

La Tabla 1 contiene también el promedio de los valores óptimos (para la función objetivo) de todos los mapas para cada algoritmo. Esto muestra que la variante GLBest es superior, como promedio, tomando como ejemplo el conjunto de 50 mapas seleccionados. Además, analizando la ejecución de los tres algoritmos (GBest, LBest, GLBest) para cada mapa, la variante GLBest supera a las demás en 48 de los 50 mapas. Estos resultados son comprobados estadísticamente a través del test no paramétrico de Friedman (Friedman 1937), donde el *p-value* obtenido sugiere que existen diferencias significativas entre los comportamientos de los algoritmos. La Tabla 2 muestra el ranking de medias obtenido por el test de Friedman, donde se observa que la variante GLBest aporta las diferencias significativas en el grupo, posicionándose en el primer lugar. Por tanto, las pruebas estadísticas sobre los valores de errores obtenidos justifican la elección de GLBest como topología para el algoritmo de optimización utilizado.

Tabla 2. Ranking de medias obtenido por Friedman.

Topología	Ranking de medias
GBest	2.64
LBest	2.32
GLBest	1.04

En la tabla siguiente se muestran, para cada variante, el número de mapas que cumplen que las cinco evaluaciones realizadas convergen a un estado atractor de equilibrio único o bien a un estado

de inestabilidad. También se muestra la cantidad de mapas convergen a un atractor de equilibrio que no es único para toda entrada, en las cinco evaluaciones realizadas. Para los demás mapas las evaluaciones son mixtas, es decir, no todas cumplen con los requisitos para ser incluidas en alguno de los dos grupos mencionados anteriormente. Se observa que el algoritmo GLBest es el que mejores resultados arroja, teniendo en cuenta que son deseables configuraciones que hagan converger los mapas a estados de equilibrio que no sean únicos para toda entrada.

Tabla 3. Análisis de estados alcanzados por las variantes de PSO.

Variante	GBest	LBest	GLBest
Estados de equilibrio Único o Inestable	5	3	3
Estados de equilibrio Diferentes	37	38	40

Las simulaciones y pruebas estadísticas descritas indican que la variante GLBest es la mejor para afrontar el problema de optimización planteado en la sección 2.2, ya sea desde el punto de vista del valor óptimo encontrado como al analizar la cantidad de mapas que convergen a estados de equilibrio no únicos. A partir de este punto, los experimentos restantes se realizan usando esta variante de PSO.

3.2.2 SELECCIÓN DEL PARÁMETRO α PARA LA FUNCIÓN OBJETIVO

Se debe seleccionar el mejor parámetro α para la función objetivo (2.4), es decir, el que permite al optimizador (GLBest) encontrar las mejores configuraciones en cuanto a convergencia y precisión. Para esto se toman los 50 mapas utilizados para el experimento anterior y, para cada valor de α (0, 0.1, 0.2, ..., 0.9, 1), se ejecutan cinco evaluaciones (en cada una se calcula el error promedio de las neuronas) del algoritmo por mapa. Además, para cada evaluación se analiza si el mapa con la configuración óptima encontrada cumple algunos requisitos para su uso satisfactorio en problemas de simulación de escenarios. Para este tipo de problemas se desea que el mapa converja a diferentes estados atractores de equilibrio, siendo contraproducentes la convergencia a un punto fijo único y la inestabilidad (esta característica no es tan perjudicial mientras se mantenga buena precisión en las salidas del mapa).

La Tabla 4 contiene, para cada mapa y valor de α , el promedio de los errores de las neuronas para las cinco evaluaciones, y además, se promedian estos resultados para cada valor de α usado.

Tabla 4. Promedio de errores de las neuronas para cada combinación.

Mapa\ α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
M1	0.337	0.403	0.413	0.426	0.411	0.408	0.352	0.420	0.432	0.442	0.454
M2	0.391	0.467	0.478	0.448	0.478	0.460	0.367	0.434	0.495	0.521	0.496
M3	0.364	0.448	0.487	0.419	0.497	0.447	0.416	0.446	0.476	0.395	0.498
M4	0.284	0.321	0.361	0.413	0.407	0.353	0.306	0.379	0.385	0.374	0.365
M5	0.089	0.113	0.094	0.101	0.107	0.104	0.116	0.111	0.120	0.120	0.119
M6	0.197	0.222	0.245	0.204	0.171	0.222	0.142	0.238	0.198	0.226	0.197
M7	0.523	0.619	0.532	0.577	0.526	0.545	0.472	0.447	0.520	0.575	0.594
M8	0.134	0.159	0.159	0.152	0.149	0.174	0.132	0.161	0.162	0.183	0.176
M9	0.202	0.213	0.237	0.203	0.223	0.246	0.204	0.195	0.167	0.226	0.272
M10	0.113	0.132	0.160	0.137	0.128	0.131	0.120	0.135	0.154	0.139	0.141
M11	0.227	0.278	0.309	0.303	0.315	0.303	0.246	0.295	0.293	0.302	0.308
M12	0.199	0.277	0.255	0.269	0.256	0.274	0.179	0.229	0.253	0.259	0.264
M13	0.090	0.104	0.105	0.099	0.109	0.104	0.079	0.101	0.107	0.108	0.106
M14	0.152	0.160	0.176	0.196	0.243	0.182	0.158	0.184	0.214	0.199	0.166
M15	0.307	0.329	0.323	0.332	0.355	0.327	0.274	0.326	0.302	0.291	0.326
M16	0.097	0.098	0.139	0.105	0.099	0.125	0.077	0.094	0.093	0.098	0.101
M17	0.192	0.174	0.184	0.190	0.176	0.171	0.162	0.169	0.193	0.178	0.174
M18	0.148	0.174	0.179	0.179	0.181	0.188	0.142	0.173	0.179	0.182	0.184
M19	0.101	0.148	0.121	0.117	0.158	0.125	0.075	0.115	0.104	0.136	0.201
M20	0.113	0.132	0.132	0.139	0.107	0.111	0.117	0.140	0.123	0.117	0.131
M21	0.095	0.131	0.154	0.140	0.129	0.138	0.115	0.112	0.119	0.142	0.126
M22	0.164	0.194	0.178	0.214	0.206	0.183	0.246	0.206	0.172	0.264	0.187
M23	0.240	0.245	0.264	0.303	0.294	0.236	0.249	0.271	0.302	0.241	0.307
M24	0.263	0.259	0.270	0.238	0.217	0.260	0.209	0.221	0.246	0.204	0.224
M25	0.179	0.352	0.338	0.274	0.255	0.168	0.182	0.291	0.241	0.296	0.267
M26	0.095	0.115	0.167	0.147	0.154	0.112	0.270	0.265	0.265	0.278	0.291
M27	0.111	0.132	0.154	0.135	0.138	0.126	0.185	0.298	0.206	0.297	0.201
M28	0.138	0.179	0.165	0.173	0.184	0.192	0.119	0.133	0.133	0.150	0.162
M29	0.168	0.212	0.218	0.229	0.218	0.213	0.129	0.128	0.142	0.154	0.141
M30	0.153	0.188	0.179	0.181	0.200	0.187	0.154	0.193	0.176	0.183	0.189
M31	0.174	0.202	0.211	0.212	0.209	0.208	0.173	0.206	0.202	0.217	0.204
M32	0.172	0.205	0.209	0.222	0.212	0.216	0.177	0.211	0.217	0.227	0.215
M33	0.136	0.177	0.184	0.180	0.172	0.175	0.148	0.181	0.182	0.190	0.180
M34	0.180	0.188	0.197	0.187	0.186	0.174	0.176	0.262	0.218	0.206	0.190
M35	0.095	0.127	0.136	0.135	0.122	0.130	0.120	0.135	0.136	0.137	0.133
M36	0.190	0.230	0.226	0.241	0.233	0.232	0.181	0.219	0.215	0.229	0.222
M37	0.162	0.205	0.197	0.209	0.200	0.212	0.160	0.198	0.203	0.196	0.207
M38	0.140	0.154	0.173	0.184	0.178	0.171	0.154	0.162	0.162	0.171	0.176
M39	0.074	0.077	0.103	0.091	0.085	0.091	0.065	0.086	0.092	0.088	0.104
M40	0.118	0.145	0.150	0.161	0.168	0.164	0.135	0.141	0.157	0.163	0.161

M41	0.173	0.202	0.222	0.191	0.208	0.183	0.148	0.235	0.198	0.195	0.216
M42	0.049	0.058	0.056	0.045	0.058	0.052	0.043	0.050	0.072	0.053	0.100
M43	0.162	0.196	0.198	0.216	0.221	0.209	0.159	0.187	0.214	0.236	0.212
M44	0.080	0.092	0.097	0.096	0.099	0.093	0.078	0.099	0.095	0.094	0.099
M45	0.252	0.328	0.317	0.304	0.330	0.300	0.271	0.291	0.300	0.308	0.301
M46	0.288	0.328	0.353	0.325	0.330	0.340	0.252	0.323	0.317	0.325	0.320
M47	0.263	0.239	0.297	0.260	0.291	0.239	0.216	0.316	0.281	0.298	0.311
M48	0.145	0.204	0.203	0.193	0.193	0.188	0.159	0.180	0.203	0.262	0.208
M49	0.214	0.245	0.261	0.290	0.255	0.283	0.209	0.256	0.260	0.307	0.287
M50	0.228	0.277	0.328	0.275	0.287	0.248	0.202	0.249	0.259	0.244	0.217
PROMEDIO	0.183	0.217	0.226	0.221	0.223	0.214	0.184	0.218	0.219	0.229	0.229

Analizando los promedios de los resultados se tiene que cuando $\alpha = 0$ y cuando $\alpha = 0.6$ se obtienen los mejores errores promedio para las neuronas de los 50 mapas involucrados en el experimento. Estos resultados son comprobados estadísticamente a través del test no paramétrico de Friedman (Friedman 1937), donde el p -value obtenido sugiere que existen diferencias significativas entre los comportamientos de los algoritmos. La Tabla 5 muestra el ranking de medias obtenido por el test de Friedman, donde se observa que la variante donde $\alpha = 0.6$ aporta las diferencias significativas en el grupo, posicionándose en el primer lugar. Por tanto, las pruebas estadísticas sobre los valores de errores obtenidos justifican la elección de $\alpha = 0.6$ como el parámetro para la función objetivo utilizada.

Tabla 5. Ranking de medias obtenido por Friedman.

Parámetro α	Ranking de medias obtenido por Friedman
0.0	2.38
0.1	6.02
0.2	7.48
0.3	7.16
0.4	7.12
0.5	6.08
0.6	2.28
0.7	5.68
0.8	6.38
0.9	7.68
1.0	7.74

En la tabla siguiente se muestran, para cada α , la cantidad de mapas para los que las cinco evaluaciones realizadas convergen a un estado atractor de equilibrio único o a un estado de inestabilidad. También aparece la cantidad de mapas para los que las cinco evaluaciones convergen a un atractor de equilibrio que no es único para toda entrada. Se observa que con $\alpha = 0.6$ y $\alpha = 0$ se obtienen los mejores resultados, teniendo en cuenta que son deseables configuraciones que hagan converger los mapas a estados de equilibrio que no sean únicos para toda entrada

Tabla 6. Análisis de estados alcanzados para los diferentes valores de α .

Mapa \ α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Estados de equilibrio Único o Inestable	5	4	5	4	4	3	4	3	5	3	5
Estados de equilibrio Diferentes	44	41	40	40	42	40	44	40	39	41	39

Los experimentos y pruebas estadísticas descritas indican que la variante con $\alpha = 0.6$ es la mejor para afrontar el problema de optimización planteado en la sección 2.2, ya sea desde el punto de vista de los promedios de errores de las neuronas como al analizar la cantidad de mapas que convergen a estados de equilibrio no únicos. Entonces queda definido $\alpha = 0.6$ como el parámetro óptimo para que el algoritmo de optimización propuesto encuentre configuraciones que mejoren la convergencia de los MCD Sigmoideos y al mismo tiempo mantengan la precisión en valores aceptables.

3.3 UTILIDAD DE LOS RESULTADOS MATEMÁTICOS

En la sección 2.4 se proponen varias definiciones que tienen estrecha relación con la convergencia y precisión de los MCD. Las Definiciones 7 y 8 son las más generales e interesantes, ya que constituyen un criterio, tanto para la estabilidad como para la precisión de un mapa. También se plantean condiciones de suficiencia que permiten reducir la complejidad computacional asociada a la comprobación directa de las definiciones antes mencionadas.

Para evaluar los resultados obtenidos, usando el algoritmo de optimización propuesto en la sección 2.2, se elaboran varios experimentos que involucran los 50 mapas usados en el epígrafe 3.2. Se

utilizan las condiciones de suficiencia antes señaladas para encontrar los conjuntos óptimos E tales que los mapas sean E -estables o E -inestables, según sea necesario. Para encontrar estos conjuntos se usa el método de optimización GLBest, sobre el cual ya se ha tratado en este trabajo.

3.3.1 ANALIZANDO E -ESTABILIDAD

Siendo $E = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N\}$, son deseables mapas E -estables con $Prom(E) = \frac{\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_N}{N}$ tan cerca de 0 como sea posible. Se puede asegurar que un mapa tiene mejor estabilidad y precisión cuando $Prom(E)$ es muy cercano a 0. Para cada mapa se minimizará $Prom(E)$ para tres variantes:

1. $\varepsilon_1 = \varepsilon_2 = \dots = \varepsilon_N$ y no se conocen la inclinación ni el desplazamiento para ninguna neurona del mapa.
2. Los valores de ε_i no son necesariamente iguales y no se conocen la inclinación ni el desplazamiento para ninguna neurona del mapa.
3. Los valores de ε_i no son necesariamente iguales y los valores óptimos para la inclinación y el desplazamiento de cada neurona son estimados por el algoritmo propuesto en la sección 2.2.

El objetivo de la optimización será minimizar este promedio de forma tal que se cumpla la condición de suficiencia 1 (para la variante 3) o la 2 (para las variantes 1 y 2) para la Definición 7. A continuación, se muestran los resultados obtenidos para un mapa del experimento aplicando las tres variantes.

Variante 1:

Cuando todos los elementos del conjunto E son iguales a 0.998 se puede asegurar que el mapa en cuestión es E -estable. Esto es análogo a decir que el mapa es 0.998-estable, según la Definición 4. Como el Corolario 4.2 plantea que todo mapa es 1-estable, el hecho de que un mapa sea 0.998-estable no aporta mucha información sobre sus propiedades de convergencia o precisión.

Variante 2:

El promedio de los elementos del conjunto E es 0.9058440884693836. Los valores óptimos encontrados para cada neurona se muestran debajo.

Tabla 7. Valores admisibles por neurona.

Neurona	Valor encontrado
1	1.0
2	1.0
3	0.9494237773639354
4	0.7447552752676795
5	1.0
6	1.0
7	0.764349282427896
8	1.0
9	1.0
10	1.0
11	1.0
12	0.6098303917136177
13	1.0

Neurona	Valor encontrado
14	0.9744288772831534
15	1.0
16	1.0
17	1.0
18	0.59807536440455
19	1.0
20	0.6442398873650801
21	1.0
22	0.4548728875476862
23	0.7276611487167571
24	0.9515300109363377
25	1.0
26	1.0
27	1.0
28	1.0

Variante 3:

El algoritmo que calcula los valores óptimos para cada λ_i y cada h_i encontró, como mejor resultado, una configuración que hace que el mapa converja a un estado atractor de equilibrio único para todas las entradas. El error promedio de las neuronas usando esta configuración es 0.4088248988584714. Usando la condición de suficiencia 1 para la Definición 7, se obtienen para cada neurona los errores admisibles siguientes:

Tabla 8. Valores admisibles por neurona.

Neurona	Valor encontrado
1	0.9422876317279055
2	1.0
3	1.0
4	1.0
5	1.0
6	1.0
7	0.9422876317279055
8	1.0
9	1.0
10	1.0
11	1.0
12	0.5685631502037399
13	0.8696720501818821

Neurona	Valor encontrado
14	1.0
15	1.0
16	1.0
17	1.0
18	0.5161674052619969
19	1.0
20	0.6400370916229358
21	0.6364222954555797
22	0.9165841483551984
23	1.0
24	1.0
25	1.0
26	1.0
27	0.9291254743530885
28	1.0

El promedio de estos errores, admisibles por cada neurona para cumplir la E -estabilidad, es 0.9271838171032225.

Para los demás mapas del experimento ocurren situaciones análogas a la observada con este mapa. Para las tres variantes los errores admisibles son muy grandes y, por tanto, los resultados no constituyen un criterio de peso para decidir si un mapa será convergente (ni siquiera con cierto margen de error) o si tendrá buena precisión.

Se puede afirmar que, analizando los resultados del experimento anterior, la tarea de encontrar el conjunto E óptimo tal que un mapa sea E -estable mediante el cumplimiento de las condiciones suficientes 1 y 2 para la Definición 7, no tiene utilidad práctica. Por tanto, mediante el análisis de E -estabilidad en los 50 mapas tomados en los experimentos de la sección 3.2, no se puede validar la calidad del algoritmo propuesto para mejorar la convergencia en MCD Sigmoideos.

3.3.2 ANALIZANDO E -INESTABILIDAD

Siendo $E = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N\}$, no tiene sentido encontrar mapas E -inestables con $Prom(E) = \frac{\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_N}{N}$ tan cerca de 1 como sea posible, ya que la E -inestabilidad puede ser causada por una única neurona (incumpliendo con su cota ε_i), y de esta forma, las demás cotas pueden tomar valor 1. Así se obtienen valores muy cercanos a 1, por lo que maximizar $Prom(E)$ en este caso no aporta la información deseada (excepto cuando todos los ε_i son iguales). Para poder usar esta maximización como un buen criterio acerca de las características de inestabilidad de un mapa debemos encontrar el conjunto E tal que el mapa sea E -inestable y que además, cada neurona incumpla con su cota asociada. Esto significa que no basta con que la neurona i -ésima incumpla con su cota asociada (ε_i), para alguna entrada, sino que toda neurona debe incumplir para al menos una entrada.

En los experimentos realizados usando la variante de maximización descrita en el párrafo anterior no se encuentran soluciones para ningún mapa, ya que es muy difícil que todas las neuronas incumplan con su cota asociada, contando con las condiciones de suficiencia 1 y 2 para la Definición 8 como únicas herramientas para comprobar la E -inestabilidad. Teniendo en cuenta esto, para cada mapa se maximizará $Prom(E)$ para dos variantes:

1. $\varepsilon_1 = \varepsilon_2 = \dots = \varepsilon_N$ y no se conocen la inclinación ni el desplazamiento para ninguna neurona del mapa.
2. $\varepsilon_1 = \varepsilon_2 = \dots = \varepsilon_N$ y los valores óptimos para la inclinación y el desplazamiento de cada neurona son estimados por el algoritmo propuesto en la sección 2.2.

El objetivo de la optimización será maximizar este promedio de forma tal que se cumpla la condición de suficiencia 1 (para la variante 2) o la 2 (para la variante 1) para la Definición 8. A continuación, se muestran los resultados obtenidos para un mapa del experimento aplicando ambas variantes.

Tabla 9. Experimento usando las condiciones de suficiencia para la E-inestabilidad.

Mapa	Variante 1	Variante 2	Promedio de error de las neuronas para configuración óptima	Estable (estados atractores de equilibrio no únicos)
M1	0.03	0.499	0.34769389	NO
M2	0.199	0.531	0.41116009	NO
M3	0.334	0.449	0.15364539	SI
M4	0.017	0.167	0.27418259	NO
M5	0.08	0.171	0.11288135	SI
M6	0.141	0.246	0.14891653	SI
M7	0.136	0.344	0.42704473	NO
M8	0.377	0.467	0.11431412	SI
M9	0.098	0.159	0.08867061	SI
M10	0.065	0.179	0.13482825	SI
M11	0.24	0.334	0.23509872	SI
M12	0.192	0.282	0.2448221	NO
M13	0.235	0.239	0.08274791	SI
M14	0.154	0.27	0.13973912	SI
M15	0.142	0.257	0.26807257	SI
M16	0.312	0.358	0.07588829	SI
M17	0.199	0.265	0.12087617	SI
M18	0.277	0.372	0.13560304	SI
M19	0.248	0.277	0.06462375	SI
M20	0.316	0.342	0.1020245	SI
M21	0.126	0.354	0.05947825	SI
M22	0.208	0.565	0.16065268	SI
M23	0.152	0.316	0.10975821	SI
M24	0.209	0.427	0.16126533	SI
M25	0.372	0.544	0.20728788	SI
M26	0.337	0.538	0.20346945	SI
M27	0.356	0.5	0.1304078	SI
M28	0.41	0.474	0.07409294	SI
M29	0.276	0.326	0.09776367	SI
M30	0.456	0.496	0.15395638	SI
M31	0.192	0.342	0.16065048	SI
M32	0.418	0.516	0.16845103	SI
M33	0.175	0.276	0.12816802	SI
M34	0.199	0.257	0.10209024	SI
M35	0.095	0.306	0.10055816	SI
M36	0.245	0.374	0.18056789	SI
M37	0.277	0.289	0.14640023	SI
M38	0.232	0.34	0.13922141	SI
M39	0.383	0.406	0.06167487	SI
M40	0.195	0.222	0.1080901	SI
M41	0.295	0.305	0.14111091	SI
M42	0.16	0.202	0.03810426	SI
M43	0.42	0.422	0.14839892	SI
M44	0.219	0.294	0.06241331	SI
M45	0.37	0.37	0.24967407	SI

M46	0.239	0.37	0.27971955	SI
M47	0.439	0.473	0.19455566	SI
M48	0.46	0.56	0.14247311	SI
M49	0.255	0.326	0.18198815	SI
M50	0.185	0.22	0.14288054	SI

Los resultados de la columna correspondiente a la Variante 1 en la Tabla 9 exponen, sin conocerse los parámetros de las funciones de transformación correspondientes a cada neurona, el mayor valor ε_i tal que el mapa i -ésimo es ε_i -inestable. Análogamente está conformada la columna para la Variante 2, pero conociéndose los parámetros de las funciones de transformación encontrados por el método de optimización propuesto en la sección 2.2. La siguiente columna muestra el error promedio de las neuronas de cada mapa, luego del proceso de optimización, mientras que la última columna contiene información sobre la estabilidad de los mapas a estados atractores de equilibrio no únicos.

Para entender mejor los datos presentados en la Tabla 9 se analiza un mapa particular. Por ejemplo, para el mapa 39 se tiene un error promedio de las neuronas igual a 0.06167487, que constituye un resultado relevante, alcanzándose esta precisión haciendo que el mapa converja a estados de equilibrio no únicos. Pero sin necesidad de realizar el proceso de inferencia ni precisar los parámetros de las funciones de transformación (corriendo el algoritmo de optimización), se puede afirmar que al menos una neurona del mapa incumplirá con su cota asociada (0.383). Luego del proceso de optimización se puede mejorar el valor anterior, ya que al menos una neurona incumplirá su nueva cota (0.406).

Para los demás mapas del experimento ocurren situaciones parecidas a la observada con este mapa. Además, al calcular los valores asociados a cada mapa para las Variantes 1 y 2, se verificó que en casi la totalidad de los experimentos solo una neurona incumple su cota asociada. Esto destaca la calidad de los resultados de la optimización, ya que se logran errores promedio bastante pequeños para las neuronas, y solo se puede asegurar cierto error de precisión para una neurona en la gran mayoría de los casos.

3.4 CONCLUSIONES PARCIALES

Las simulaciones y pruebas estadísticas realizadas sobre el conjunto de mapas generado, indican que la variante GLBest es la mejor para afrontar el problema de optimización planteado en la sección 2.2. Resultados numéricos como los valores óptimos encontrados y la convergencia, evidencian su superioridad respecto al PSO con topología global (GBest) y al PSO con topología de anillo (LBest).

Los experimentos también se orientaron a encontrar el parámetro α para la función objetivo (2.4) con el que se obtienen los mejores resultados. El test de Friedman resultó bastante útil para escoger este parámetro, ya que para $\alpha = 0.6$ y $\alpha = 0$, la precisión de los MCD y los totales de mapas que convergieron a puntos de equilibrio no únicos, fueron cercanos. Finalmente queda definido $\alpha = 0.6$ como el parámetro óptimo para que el algoritmo de optimización propuesto encuentre configuraciones que mejoren la convergencia de los MCD Sigmoideos y al mismo tiempo mantengan la precisión en valores aceptables.

La aplicación de las condiciones de suficiencia propuestas en la sección 2.4, para el caso de la E -estabilidad, no resultó satisfactoria. No se puede validar la calidad del algoritmo propuesto para mejorar la convergencia en MCD Sigmoideos. De otro modo, para el caso de la E -inestabilidad, los resultados alcanzados evidencian la dificultad del proceso para mejorar la convergencia sin afectar precisión. Sin necesidad de desencadenar el proceso de inferencia en un MCD Sigmoideo, se puede calcular cierto margen de error que al menos una neurona incumplirá, sin importar los parámetros inclinación (λ_i) y desplazamiento (h_i) que el método de optimización pueda encontrar. Aun calculando un nuevo margen de error sobre el mapa optimizado, se observa que hay limitaciones insuperables respecto a la precisión.

CONCLUSIONES

Aunque en los últimos años son muchos los esfuerzos, el problema de la estabilidad en MCD Sigmoideos resulta un problema de investigación abierto. Los enfoques analíticos adoptados son de poca usabilidad, ya que aseguran condiciones para que el sistema converja a un punto único. No obstante, en este trabajo se presentó un algoritmo de aprendizaje supervisado basado en Inteligencia Colectiva que permite mejorar las propiedades de convergencia de los MCD Sigmoideos preservando la precisión, en problemas de simulación de escenarios.

Esencialmente, el algoritmo consiste en estimar la inclinación y el desplazamiento de cada función sigmoidea asociada a cada neurona del mapa, en lugar de usar la misma función de umbral para todas las neuronas; con el objetivo de minimizar una función objetivo representativa en cuanto a convergencia y precisión se refiere. Esto permite mejorar las propiedades de convergencia del mapa, a la vez que se preserva la precisión, aunque no hay seguridad en encontrar un punto a atractor de equilibrio.

En principio es posible estimar la inclinación y desplazamiento adecuados de cada función sigmoidea con cualquier metaheurística poblacional, incluso usando técnicas de programación matemática. Sin embargo, los métodos de búsqueda poblacionales son capaces de encontrar soluciones cercanas al óptimo en un tiempo de ejecución razonable. GLBest (variante de PSO) combina las capacidades de convergencia de la topología global con la preservación de diversidad presente en las topologías locales, y resulta una alternativa factible para tratar el problema en cuestión. Esta hipótesis fue demostrada estadísticamente para 50 mapas con diversas topologías y porcentajes de conectividad.

Se generó un conjunto de MCD artificiales con diferentes características estructurales, sobre los cuales, las simulaciones y pruebas numéricas mostraron que el algoritmo es capaz de mejorar la convergencia del mapa, preservando su precisión. Además, usando varias definiciones y condiciones de suficiencia propuestas, se observan las dificultades superadas por el método de optimización .

RECOMENDACIONES

1. Evaluar el rendimiento del algoritmo de aprendizaje propuesto para casos de estudio reales.
2. Permitir que el método de optimización varíe la función usada para la normalización (sigmoidea, bivalente, trivalente, seno hiperbólico y coseno hiperbólico).
3. Extender los resultados matemáticos propuestos a problemas de clasificación de patrones y problemas de decisión.

REFERENCIAS BIBLIOGRÁFICAS

- Axelrod, R., 1976. *Structure of Decision: The Cognitive Maps of political Elites*, Princeton University Press.
- Boutalis, Y., Kottas, T.L. & Christodoulou, M.C., 2009. Adaptive Estimation of Fuzzy Cognitive Maps With Proven Stability and Parameter Convergence. *IEEE Transactions on Fuzzy Systems*, 17, pp.874–889.
- Bratton, D. & Kennedy, J., 2007. Defining a Standard for Particle Swarm Optimization. In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*. pp. 120–127.
- Bueno, S. & Salmeron, J.L., 2009. Benchmarking main activation functions in fuzzy cognitive maps. *Expert Systems with Applications*, 36(3 PART 1), pp.5221–5229. Available at: <http://dx.doi.org/10.1016/j.eswa.2008.06.072>.
- Carvalho, J.P. & Tomé, J.A.B., 2002. Issues on the Stability of Fuzzy Cognitive Maps and Rule-Based Fuzzy Cognitive João Paulo Carvalho. *Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American*, pp.105–110.
- Chen, Y., Mazlack, L.J. & Lu, L.J., 2012. Learning fuzzy cognitive maps from data by ant colony optimization. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*. ACM, pp. 9–16.
- Clerc, M. & Kennedy, J., 2002. The Particle Swarm — Explosion , Stability , and Convergence in a Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, 6(1), pp.58–73.
- Codara, L., 1998. *Le Mapped Cognitive*. Carrocci.
- Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200), pp.675–701.
- Huerga, A.V., 2002. A balanced differential learning algorithm in fuzzy cognitive maps. *Proc. 16th International Workshop on Qualitative Reasoning*, pp.1–7.
- Kennedy, J. & Eberhart, R., 1995. Particle Swarm Optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*. pp. 1942–1948.
- Kennedy, J. & Mendes, R., 2006. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics*, 36, pp.515–519.
- Knight, C.J.K., Lloyd, D.J.B., Penn, A.S., et al., 2014. Linear and sigmoidal fuzzy cognitive maps: An analysis of fixed points. *Applied Soft Computing Journal*, 15, pp.193–202. Available at: <http://dx.doi.org/10.1016/j.asoc.2013.10.030>.
- Knight, C.J.K., Lloyd, D.J.B. & Penn, A.S., 2014. Linear and sigmoidal fuzzy cognitive maps: An analysis of fixed points. *Applied Soft Computing Journal*, 15, pp.193–202. Available at: <http://dx.doi.org/10.1016/j.asoc.2013.10.030>.
- Kosko, B., 1984. Fuzzy Cognitive Maps. *International Journal of Approximate Reasoning*, pp.377–393.
- Kosko, B., 1986. Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, 24(1), pp.65–75. Available at: <http://www.sciencedirect.com/science/article/pii/S0020737386800402>.
- Kosko, B., 1998. Global stability of generalized additive fuzzy systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 28(3), pp.441–452. Available at:

- http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=704584.
- Kosko, B., 1988. Hidden patterns in combined and adaptive knowledge networks. *International Journal of Approximate Reasoning*, 2(4), pp.377–393. Available at: <http://www.sciencedirect.com/science/article/pii/0888613X88901119>.
- Kosko, B., 1992. *Neural Networks and Fuzzy systems, a dynamic system approach to machine intelligence*, Prentice-Hall, Englewood Cliffs.
- Kottas, T.L., Boutalis, Y.S. & Christodoulou, M.A., 2012. Bi-linear adaptive estimation of Fuzzy Cognitive Networks. *Applied Soft Computing*, 12(12), pp.3736–3756. Available at: <http://dx.doi.org/10.1016/j.asoc.2012.01.025>.
- Koulouriotis, D.E., Diakoulakis, I.E. & M., E.D., 2001. Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior. In *IEEE Congress on Evolutionary Computation*. pp. 364–371.
- Li, X., 2010. Niching Without Niching Parameters: Particle Swarm Optimization Using a Ring Topology. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 14, pp.150–169.
- Nápoles, G., 2014. *Algoritmo para mejorar la convergencia en Mapas Cognitivos Difusos Sigmoidales*.
- Nápoles, G. et al., 2013. Learning of fuzzy cognitive maps for simulation and knowledge discovery. In *Studies on Knowledge Discovery, Knowledge Management and Decision Making, EUREKA 2013*. pp. 27–36.
- Nápoles, G., Grau, I., et al., 2014. Towards Swarm Diversity : Random Sampling in Variable Neighborhoods Procedure Using a Levy Distribution. *Computación y Sistemas*, 18(1), pp.79–95.
- Nápoles, G., Bello, R. & Vanhoof, K., 2014. Global stability improvements on adjusted Sigmoid Fuzzy Cognitive Maps. *Intelligent Data Analysis*.
- Nápoles, G., Grau, I. & Bello, R., 2012. Constricted Particle Swarm Optimization based algorithm for global optimization. *Polibits*, 46(46), pp.5–11.
- Papageorgiou, E.I. & Salmeron, J.L., 2013. A review of fuzzy cognitive maps research during the last decade. *IEEE Transactions on Fuzzy Systems*, 21(1), pp.66–79.
- Papageorgiou, E.I., Stylios, C.D. & Groumpos, P.P., 2004. Active Hebbian learning algorithm to train fuzzy cognitive maps. *International Journal of Approximate Reasoning*, 37, pp.219–249.
- Papageorgiou, E.I., Stylios, C.D. & Groumpos, P.P., 2003. Fuzzy cognitive map learning based on nonlinear Hebbian rule. *Australian Conference on Artificial Intelligence*, pp.256–268.
- Papageorgiou, E.I., 2012. Learning Algorithms for Fuzzy Cognitive Maps - A Review Study. *IEEE Transaction on Systems, Man, and Cybernetics*, 42(2), pp.150–163.
- Papakostas, G.A. et al., 2008. Fuzzy Cognitive Maps for Pattern Recognition Applications. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(8), pp.1461–1486.
- Papakostas, G.A. et al., 2012. Towards Hebbian learning of Fuzzy Cognitive Maps in pattern classification problems. *Expert Systems With Applications*, 39(12), pp.10620–10629. Available at: <http://dx.doi.org/10.1016/j.eswa.2012.02.148>.
- Parsopoulos, K.E. et al., 2003. A first study of fuzzy cognitive maps learning using particle swarm optimization. In *IEEE Congress on Evolutionary Computation*. pp. 1440–1447.
- Poli, R., Kennedy, J. & Blackwell, T., 2007. Particle Swarm Optimization – An overview. *IEEE Trans. on Evol. Computation*, 1, pp.37–57.
- Rudin, W., 1964. *Principles of Mathematical Analysis*. New York: McGraw-Hill.

- Shi, Y. & Eberhart, R., 1998. A Modified Particle Swarm Optimizer. In *IEEE Congress on Evolutionary Computation*. pp. 69–73.
- Stach, W. et al., 2005. Genetic learning of fuzzy cognitive maps. *Fuzzy Sets and Systems*, 153, pp.371–401.
- Stach, W., Kurgan, L. & Pedrycz, W., 2008. Data driven nonlinear Hebbian learning method for fuzzy cognitive maps. In *IEEE World Congress Computation Intelligence*. Hong Kong: IEEE.
- Tsadiras, A.K., 2008. Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. *Information Sciences*, 178(20), pp.3880–3894.
- Wang, L., Pichler, E.E. & Ross, J., 1990. Oscillations and chaos in neural networks: An exactly solvable mode. *PNAS*, 87, pp.9467–9471.
- Wang, Y. et al., 2011. Self-adaptive learning based particle swarm optimization. *Information Sciences*, 181(20), pp.4515–4538. Available at: <http://dx.doi.org/10.1016/j.ins.2010.07.013>.