OPTIMIZACIÓN DE LA ASIGNACIÓN DE CANALES UTILIZANDO REDES NEURONALES DE TIPO *HOPFIELD*

CLAUDIA PATRICIA CAMACHO OBREGÓN
VÍCTOR MANUEL RUIZ HERRERA

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C.
NOVIEMBRE DE 2010

OPTIMIZACIÓN DE LA ASIGNACIÓN DE CANALES UTILIZANDO REDES NEURONALES DE TIPO *HOPFIELD*

CLAUDIA PATRICIA CAMACHO OBREGÓN
VÍCTOR MANUEL RUIZ HERRERA

Trabajo de grado para optar por el título de
Ingeniero Electrónico

Director:
CARLOS IVÁN PÁEZ RUEDA
Ingeniero Electrónico, MSc.

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C.
NOVIEMBRE DE 2010

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE ELECTRÓNICA

RECTOR MAGNÍFICO:	**JOAQUÍN SÁNCHEZ GARCÍA S. J.**
DECANO ACADÉMICO:	**Ing. FRANCISCO J. REBOLLEDO M.**
DECANO DEL MEDIO UNIVERSITARIO:	**SERGIO BERNAL RESTREPO S. J.**
DIRECTOR DE CARRERA:	**Ing. JUAN MANUEL CRUZ.**
DIRECTOR DEL TRABAJO DE GRADO:	**Ing. CARLOS IVÁN PÁEZ RUEDA**

Nota de aceptación

_____

_____

_____

_____

Presidente del Jurado

_____

Jurado

_____

Jurado

Bogotá 22 Noviembre 2010

ARTÍCULO 23 DE LA RESOLUCIÓN N° 13 DE JUNIO DE 1946

"La universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus trabajos de grado. Solo velara porque no se publique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien que se vea en ellos el anhelo de buscar la verdad y la justicia"

DEDICATORIA

Principalmente a Dios por estar conmigo en todo momento,
A mi familia, por ser el gran apoyo que siempre necesité.
También a mis compañeros y amigos durante la carrera,
Y especialmente a Víctor quien hizo parte de este gran proceso.
A todos, con mucho cariño. Eternamente agradecida.
Claudia

Con mucho amor, a mis padres, a quienes les debo todo
Y con quienes estaré eternamente agradecido.
A Claudia, quien además de ser parte integral de este proyecto,
me dio fuerzas y ánimo durante circunstancias adversas.
Víctor

AGRADECIMIENTOS

A nuestro director, Carlos Iván Páez Rueda por su gran orientación y realimentación durante las etapas de este proyecto. Fue una gran ayuda, con disposición incondicional y paciencia ilimitada. Especialmente por haber confiado en nosotros para vincularnos en este gran desafío.

Al ingeniero José Luis Uribe, y todo su equipo del laboratorio de la facultad de ingeniería electrónica por facilitarnos con gran amabilidad los medios técnicos para el desarrollo del proyecto, por estar siempre dispuestos a ser de gran ayuda y más en los momentos críticos.

A nuestras familias y amigos, cuyo apoyo incondicional fue siempre motor de ánimo y nuevas ideas a lo largo de este trabajo. Quienes nunca perdieron la fe, que con su amor, paciencia y valor; lograron que fuéramos parte de los grandes ingenieros que construirán futuro de Colombia y el mundo.

Por último, pero lo más importante. A Dios por iluminarnos y darnos la fuerza que esta empresa requirió, aún en los momentos difíciles pudimos salir siempre adelante gracias a él. Gracias a él están nuestras familias y las demás personas anteriormente mencionadas, que hicieron posible la finalización de este proyecto con gran satisfacción.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE DIAGRAMAS

1. INTRODUCCIÓN

En la industria de las telecomunicaciones se han venido desarrollando tecnologías que hacen uso extensivo del espectro electromagnético. Debido a que este recurso es limitado y su demanda ha venido creciendo rápidamente se hace fundamental optimizar la utilización de los canales de transmisión.

Una red de comunicaciones dispone únicamente de un conjunto finito de canales, razón por la cual si se desea establecer un número considerable de conexiones es imperativo reutilizar un mismo canal para diferentes conexiones simultáneamente. Ésta reutilización de canales debe regirse por un conjunto de restricciones que aseguren el flujo confiable de información.

El problema de asignación de canales (CAP) busca resolver esta limitación mediante un método capaz de asegurar la asignación de canales de entre un conjunto finito de los mismos de forma eficiente. La asignación de canales debe asegurar que sea posible la transmisión de información entre un emisor y un receptor de forma unidireccional o bidireccional, buscando usar la mínima cantidad de recursos para cumplir con la mayor demanda posible y minimizar o evitar cualquier tipo de interferencia.

Las redes neuronales son una técnica heurística con mucho potencial para resolver el CAP. Su ventaja radica en que una red de actualización paralela puede ser implementada en ordenador secuencial o en una grilla de ordenadores. A su vez, la implementación en *hardware* de las redes neuronales es bastante simple dado que es un conjunto de unidades de proceso sencillas.

Los diferentes modelos de CAP que pueden plantearse tienen su origen en el tipo de restricciones que se imponen a la asignación a la asignación de canales, lo que es equivalente al objetivo de optimización. Estas restricciones obedecen a las necesidades del usuario a quien va dirigido el servicio. Existen dos tipos generales de asignación, la estática que soluciona el problema para una demanda de canales fija y la dinámica que lo hace para una demanda no fija de canales.

Este trabajo de grado trata el problema de asignación fija de mínimo orden, el cual será resuelto por tres métodos basados en redes neuronales. El primero método plantea una red de *Hopfield* clásica discreta, El segundo método se basa en el proceso de recocido simulado SA combinado con una red de *Hopfield.* El tercer y último método utiliza una red neuronal de tipo estocástico conocida como máquina de *Boltzmann*. Como antesala a la optimización del problema de asignación se unifican bajo un formato matricial uniforme un conjunto de *Benchmarks* conocidos internacionalmente.

Se conocen hasta ahora resultados para la red de Hopfield clásica [7], algoritmos de SA puros [14] y algoritmos de SA combinados con redes neuronales con un enfoque diferente al propuesto en este artículo [12]. La solución del CAP con métodos de MB no tiene antecedentes hasta donde conocen los autores.

2. MARCO TEÓRICO

2.1. Redes Neuronales

Es claro que el cerebro humano es superior a un computador digital en tareas diferentes a la aritmética como reconocimiento de información visual, entre otras. Entre las características del cerebro deseables en sistemas artificiales se encuentran:

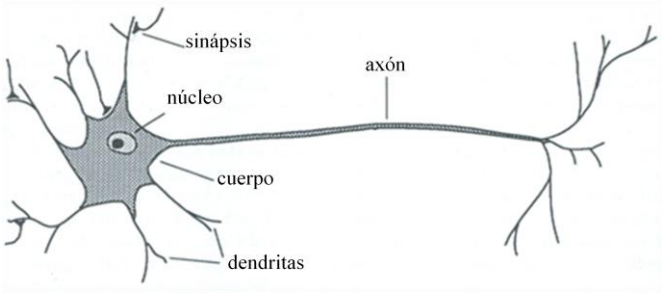
- Robustez y tolerancia a fallos: neuronas mueren a diario sin afectar significativamente el rendimiento del cerebro.
- Flexibilidad: puede ajustarse fácilmente a diferentes entornos por medio del "aprendizaje".
- Puede lidiar con información difusa, probabilística o inconsistente.
- Es altamente paralelo.
- Es pequeño, compacto, y disipa poca potencia.

En las pasadas décadas la investigación en computación neuronal estuvo fuertemente motivada por la posibilidad de crear redes computacionales artificiales para modelar sistemas artificiales que exhibieran las características previamente enunciadas. Aunque los modelos de unidades de procesamiento son muy precarios vistos desde un punto de vista neurológico, son aún útiles para adquirir información acerca de la "computación" biológica. La información consignada en esta sección es una traducción libre de [1].

2.1.1. Neuronas

Al igual que en el cerebro la unidad fundamental de las redes neuronales es llamada neurona. En el caso biológico esta unidad consta de redes en forma de árbol llamadas dendritas, conectadas al cuerpo de la neurona o soma que contiene el núcleo de la misma. Extendiéndose desde el cuerpo de la célula hay una fibra larga sencilla que eventualmente se ramifica, llamada axón. Figura 1.

Figura 1. Neurona biológica.



Fuente: Modificado por los autores de [1], pág. 2.

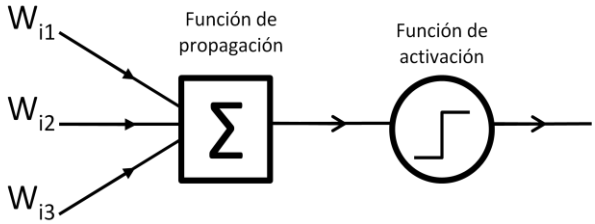
La transmisión de una señal de una neurona a otra a través de sus uniones es un proceso químico complejo cuyo efecto es el incremento o decremento de un potencial eléctrico en el cuerpo de la neurona, que recibe estímulo de unidades cercanas. Si el potencial alcanza un umbral un pulso o potencial de acción de magnitud y duración fija se envía al axón. Este impulso se conoce como "disparo" de la neurona.

McCulloch y Pitts en 1946 propusieron un modelo simple de neurona como una unidad de umbral binaria, como se muestra en la Figura 2. Este modelo computa una suma ponderada de entradas provenientes de otras unidades (función de propagación), y tiene salida de cero o uno dependiendo de si esta suma está por arriba o por debajo de cierto umbral (función de activación), siguiendo la expresión (1)**.**

$$\boldsymbol{n_i(t+1) = \Theta\left(\sum_j w_{ij} * n_j(t) - \mu_i\right)}$$

(1)

Figura 2. Neurona según modelo de McCulloch-Pitts.



Fuente: Modificado por los autores de [1], pág. 3.

En esta representación, n_i representa el estado de la unidad i, siendo este 1 cuando está se está "disparada" y 0 cuando no lo está. w_{ij} representa la fuerza de la conexión entre las neuronas i y j; dicha fuerza será positiva, en cuyo caso se llamará excitatoria, o negativa, considerándose entonces inhibitoria, t es el valor de tiempo, discreto para un modelo computacional, μ_i es el umbral de disparo de la unidad i, y $\Theta(x)$ es la función escalón unitario, dada por (2). Esta información ha sido una traducción libre de [1].

$$\boldsymbol{\Theta(x) = \begin{cases} 1\ para\ x \geq 0 \\ 0\ para\ el\ resto \end{cases}}$$

(2)

2.2. Redes de Hopfield

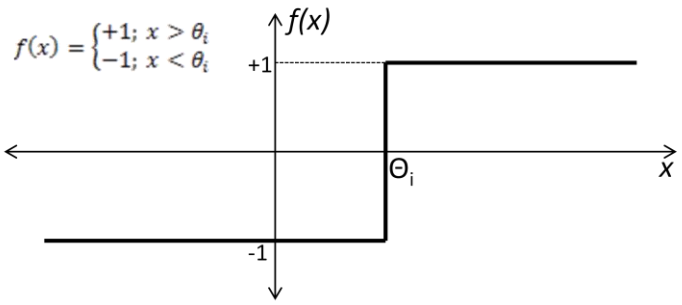
El problema de la memoria asociativa es el problema central del campo de la computación neuronal. En respuesta a este problema John Hopfield propuso un modelo de memoria asociativa llamado la red neuronal Hopfield. Información tomada de [3] y traducción libre de [1].

2.2.1. Arquitectura

En primera instancia Hopfield desarrolló el modelo de manera discreta (DH) el cual consiste en una red monocapa con un numero *N* de neuronas cuyos valores de salida son binarios representados en 0/1 ó -1/1 y una función de activación de tipo escalón. Luego de derivar el modelo DH introdujo un modelo de red continua (CH) cuyos valores de salida toman valores reales entre -1 y 1, y su función de activación es de tipo sigmoidea. Cada neurona de la red está conectada a todas las demás pero no consigo misma y los pesos asociados a las conexiones entre pares de neuronas son simétricos. Para nuestro caso al desarrollar los algoritmos usaremos redes de Hopfield en su versión discreta.

2.2.2. Función de activación

Figura 3. Función de activación.



Fuente: Modificado por los autores de [2]

Se define un valor θ_i llamado umbral de disparo de la neurona. Cuando el valor de entrada coincide exactamente con este valor la salida de la neurona *i* mantiene su valor anterior. Si la entrada de la neurona es superior o inferior a dicho umbral la salida de la misma será 1 o -1 respectivamente.

2.2.3. Funcionamiento

Se trata de una red auto-asociativa. Varias informaciones (patrones) pueden ser almacenadas en la red como si se tratase de una memoria durante la etapa de aprendizaje. Si se presenta a la entrada alguna de las informaciones almacenadas la red evoluciona hasta estabilizarse, ofreciendo entonces a la salida la información almacenada que coincide con la entrada. Si la información de entrada no coincide con ninguna de las solucionadas la red evoluciona generando como salida el patrón almacenado más parecido.

La información que recibe la red debe haber sido previamente codificada y representada en forma de vector con tantas componentes como neuronas (N). Dicha información se aplica directamente a la única capa de que consta la red (cada neurona recibe un elemento del vector de entrada). En principio una neurona recibiría como entrada las salidas de cada una de las otras neuronas, que inicialmente coincidirían con los valores de entrada multiplicadas por los pesos de las conexiones. La suma de todos estos valores es la entrada neta de la neurona a la que será aplicada la función de activación y de transferencia. El proceso anterior se repite iterativamente hasta que las salidas de las neuronas se estabilizan, lo que ocurrirá cuando dejen de cambiar de valor. El conjunto de estos N valores de salida de todas las neuronas corresponderá con alguna de las informaciones almacenadas en la etapa de aprendizaje.

2.3. Problema de Asignación de Canales (CAP)

El problema de asignación de canales CAP es la selección de canales de entre un conjunto finito de tal modo que se cumpla una demanda de tráfico sujeta a restricciones electromagnéticas. Este problema surge de la disponibilidad limitada de canales en el espectro de las comunicaciones inalámbricas lo que obliga al re-uso de estas por medio de múltiples transmisores en una misma red.

A un conjunto de conexiones de comunicación inalámbrica se le deben asignar canales de tal manera que sea posible la trasmisión de datos de una forma fiable entre el transmisor y el receptor para cada conexión. Dicha selección de canales debe tener presente como objetivo que se evite o minimice cualquier tipo de interferencia. La interferencia se mide por la relación señal a ruido en el receptor de la conexión así como se ve en la ecuación (3), donde P es la Potencia del transmisor, d es la distancia entre receptor y transmisor, y δ es un factor de desvanecimiento. En el receptor la señal proveniente del transmisor debe ser comprensible.

$$Int = \frac{P}{d^{\delta}}$$

(3)

Existen dos condiciones para que exista interferencia entre un par de canales. La primera de ellas es que las frecuencias de los canales deben estar cercanas en el espectro electromagnético o ser armónicas entre sí. La segunda condición es que estén cerca geográficamente y tengan un nivel de energía comparable en la posición del receptor.

2.3.1. Modelos de CAP.

Cada uno de los modelos depende y difiere en las restricciones electromagnéticas que se imponen en la asignación de canales. A grandes rasgos se pueden distinguir cuatro modelos principales para resolver CAP dependiendo del objetivo de la optimización. [6]

Se entiende por $|f(v)|$ la solución de la asignación para la conexión *v*, (*v*,*w)* corresponde a un par de conexiones de la red, c_v es la demanda de canales, D_v el conjunto disponible de canales para la conexión *v*, $T_{v,w}$ el conjunto de distancias permitidas entre los canales asignados a una conexión, $d_{v,w}$ es la distancia entre los canales asignados a las conexiones (*v*, *w*), P_{vwfg} penalidad por interferencia asociada a la escogencia de los canales *f* y *g* para las conexiones *v* y *w* respectivamente y k es el límite máximo de las restricciones. Es necesaria la definición de límites inferiores para los problemas de tal forma que sea posible evaluar la calidad de las soluciones. CAP es un problema NP-completo. Generalmente los métodos más sofisticados y que encuentran las soluciones óptimas en un mayor número de instancias son prácticamente aplicables para redes pequeñas.

- **MO-CAP: Asignación de mínimo orden.**

Se asignan canales de tal manera que no haya interferencia inaceptable; el objetivo es minimizar el número total de canales asignados a las conexiones. La asignación debe ser una función que cumpla las siguientes condiciones (4):

$$|f(v)| = c_v. \quad \text{(a)}$$

$$f(v) \leq D_v. \quad \text{(b)}$$

$$|\bar{f} - \bar{g}| \notin T_{v,w} \text{ para todo } \{v,w\} \in E, \bar{f} \epsilon f(v), \bar{g} \epsilon f(w), v \neq w \; o \; \bar{f} \neq \bar{g}. \quad \text{(c)}$$

$$|U_{v \epsilon V} f(v)| \leq k. \quad \text{(d)}$$

(4)

- **MS-CAP: Asignación de mínima expansión *(Span)*.**

El objetivo es minimizar la distancia entre los canales de mayor y menor frecuencia. Para ello se debe tener una función que cumpla las siguientes condiciones (5):

$$|f(v)| = c_v. \quad \text{(a)}$$

$$f(v) \subseteq D_v. \quad \text{(b)}$$

$$|\bar{f} - \bar{g}| \notin T_{v,w} \text{ Para todo } \{v,w\} \in E, \bar{f} \epsilon f(v), \bar{g} \epsilon f(w), v \neq w \; o \; \bar{f} \neq \bar{g}. \quad \text{(c)}$$

$$\max U_{v \epsilon V} f(v) - \min U_{v \epsilon V} f(v) \leq k. \quad \text{(d)}$$

(5)

- **MB-CAP: Asignación de mínimo bloqueo.**

Minimizar la probabilidad de no asignar un canal requerido por una conexión. En caso de que todas las posibles soluciones de canales tengan interferencia inaceptable se decide hacer una asignación parcial que minimiza la probabilidad de bloqueo total. Se entiende por Probabilidad de bloqueo la posibilidad de rechazar la asignación de un canal a una conexión que lo demanda. Se debe encontrar una función que cumpla las siguientes condiciones (6):

$$|f(v)| \leq c_v. \quad \text{(a)}$$

$$f(v) \subseteq D_v. \quad \text{(b)}$$

$$|\bar{f} - \bar{g}| \notin T_{v,w} \text{ Para todo } \{v,w\} \in E, \bar{f} \epsilon f(v), \bar{g} \epsilon f(w), v \neq w \; o \; \bar{f} \neq \bar{g}. \quad \text{(c)}$$

$$\sum_{v \epsilon V} b(|f(v)|) \leq k. \quad \text{(d)}$$

(6)

- **MI-CAP: Asignación de mínima interferencia.**

El objetivo es minimizar la interferencia de todas las conexiones. Asignar los canales de un conjunto limitado de tal manera que la suma total ponderada de interferencia es minimizada, para ello se debe buscar una función que cumpla las siguientes condiciones (7):

$$|f(v)| = c_v \cdot \quad \text{(a)}$$

$$f(v) \subseteq D_v \cdot \quad \text{(b)}$$

$$\sum_{\{v,w\} \epsilon E} \sum_{\substack{\bar{f} \epsilon f(v), \bar{g} \epsilon g(v) \\ (v \neq w) \vee (\bar{f} \neq \bar{g})}} P_{vw\bar{f}\bar{g}} \rho(|\bar{f} - \bar{g}| \in T_{v,w}) \leq k \cdot \quad \text{(c)}$$

(7)

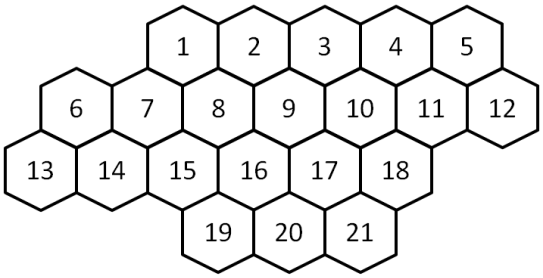
2.4. Benchmarking

2.4.1. Conjunto de instancias

Existen gran cantidad de conjuntos de instancias, entre las cuales tenemos dentro de los más usados los siguientes: Philadelphia, Cost 259, Cselt, Ratio y Televisión, entre otros. Para nuestro caso utilizamos Philadelphia el cual se explica con detalle a continuación:

Según [8] Philadelphia fue creado como conjuntos de instancias en el año 1973. Está caracterizada por 21 hexágonos (Figura 4) donde son mostradas las celdas de una red telefónica celular alrededor de Philadelphia. Recientemente se están haciendo prácticas en redes telefónicas inalámbricas como sistemas de células hexagonales. Para cada celda se tiene una demanda c_v determinada; en el modelo básico, la interferencia entre células se debe a la reutilización de canales entre células con distancia menor a d, este problema es generalizado con la matriz de distancias donde d_{vw} es la distancia entre celdas.

Figura 4. Estructura de la red.



Fuente: Modificado por los autores de [8]

2.4.2. Matrices de Compatibilidad y Vectores de Demanda

Se desarrollaran las matrices de compatibilidad y vectores de demanda en archivos .txt, por medio de Matlab para cada una de las instancias para así poder usarlas en cada uno de nuestros algoritmos. Se crearan: Para el primer caso de Funabiki F1[7] MComp4Celdas=[5,4,0,0;4,5,0,1;0,0,5,2;0,1,2,5] y Vdem4Celdas=[1,1,1,3], para las 19 instancias que contiene Philadelphia (P1…P19) [8] y P20 [14], usaremos la notación de los archivos como: MCompP1… MCompP20 para las matrices de compatibilidad y de igual forma para los vectores de demanda (Vdem). Ver código en anexos.

Partimos de (Tabla 1) para la generación de los componentes anteriormente mencionados, para establecer y clasificar cada una de las instancias.

Tabla 1. Instancias

Instancia	Matriz de Compatibilidad	Vector de demanda	Instancia	Matriz de Compatibilidad	Vector de demanda
P1	C1	D1	P11	C5	D3
P2		D2	P12	C6	D3
P3		D3	P13	C7	D2
P4		D4	P14		D3
P5		D5	P15	C8	D2
P6	C2	D1	P16		D3
P7		D2	P17	C9	D2
P8		D3	P18		D3
P9	C3	D3	P19	C10	D2
P10	C4	D3	P20	C11	D6

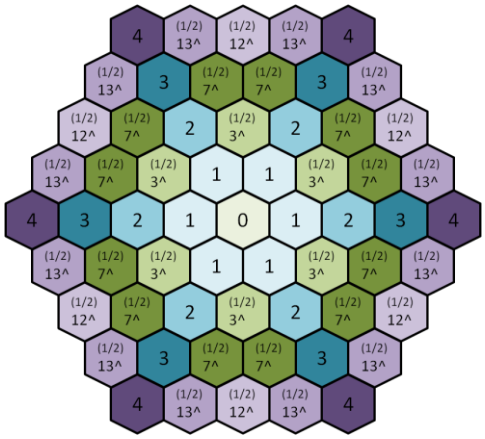
Fuente: Modificado por los autores.

Desarrollado por Carlos Páez. Derechos de autos cedidos para causas del proyecto.

Para iniciar el proceso de la creación de las matrices de compatibilidad y los vectores de demanda, lo primero que se debe tener es la matriz de distancias, luego se crearan las matrices de compatibilidad para cada instancia, dichas matrices son guardadas en .txt para poder usarlas en los demás algoritmos, y por último el ingreso de los vectores para también tenerlos en .txt.

Matriz de distancias: Es generada a partir de la estructura de la red (Figura 4), y las distancias según la posición de cada celda, como se muestra en (Figura 5).

Figura 5. Distancia entre celdas

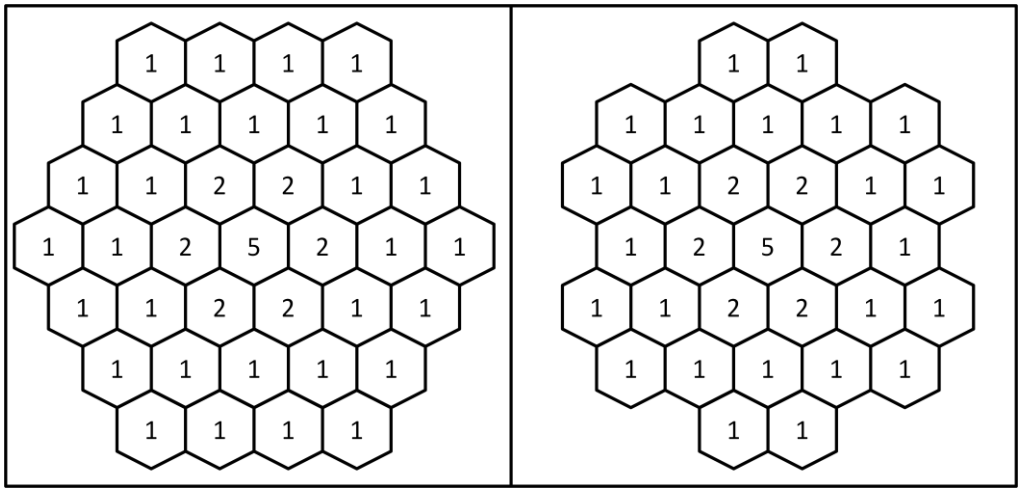


Fuente: Modificado por los autores.

Desarrollado por Carlos Páez. Derechos de autos cedidos para causas del proyecto.

Matrices de Compatibilidad: Su valor depende de dos factores, el primero de la distancia de las celdas y el segundo de la instancia, con estos datos se escoge el número que corresponda en (Figura 6) y con dicha información crear archivos .txt cuyos nombres son McompP1…Mcomp20.

Figura 6. Distancia reutilización Philadelphia



(a) P1, P3, P5, P7, P9 (b) P2, P4, P6

Fuente: Modificado de [8]

Vector de demanda: Ya sabiendo los valores de dichos vectores (Tabla 2), en Matlab lo que se hizo fue ingresarlos manualmente, y con esto, sabiendo a cual instancia corresponde cada uno cuadrarlos en archivos .txt con nombres VdemP1… VdemaP20.

Tabla 2. Vectores de demanda

VECTORES DE DEMANDA	
D1	{20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20}
D2	{5,5,5,8,12,25,30,25,30,40,40,45,20,30,25,15,15,30,20,20,25}
D3	{8,25,8,8,8,15,18,52,77,28,13,15,31,15,36,57,28,8,10,13,8}
D4	2*D3
D5	4*D3
D6	{10,11,9,5,9,4,5,7,4,8,8,9,10,7,7,6,4,5,5,7,6,4,5,7,5}

Fuente: Modificada por los autores de [8]

2.5. Simulated Annealing (SA)

Este algoritmo ha sido ampliamente aplicado para solucionar problemas de tipo combinatorio. La idea viene de una analogía del proceso físico de recocido de sólidos y la necesidad de solucionar grandes problemas de optimización combinatoria[1], a este proceso se le debe el nombre del algoritmo.

El proceso de recocido es el calentamiento de una sustancia por medio de un "baño de calor" hasta alcanzar un valor alto de temperatura en donde sus partículas solidas entran en fase liquida de manera aleatoria. Posteriormente se reduce lentamente la temperatura de la sustancia hasta obtener una estructura cristalina fuerte, en donde las partículas alcanzan un nivel bajo de energía, dado un nivel suficientemente

[1] Optimización combinatoria: Cuando la función de costo se define en un dominio discreto. En este caso, las variables de diseño son discretas y el azar de los movimientos corresponden a permutaciones en una lista de posibles movimientos.

alto de temperatura inicial y un esquema de enfriamiento suficientemente lento. Información de la sección 2.5, traducción libre de [13].

2.5.1. Proceso Físico (*Annealing Process*)

Empezando en el máximo valor de temperatura, la fase de enfriamiento del proceso puede describirse de la siguiente manera: Para cada valor de temperatura el sólido puede alcanzar el equilibrio térmico, caracterizado por una probabilidad de estar en un estado con energía E dado por la distribución de Boltzmann (8); donde Z(T) es el Factor de Normalización, K_B la Constante de Boltzmann y $-\frac{E}{K_B T}$: el factor de Boltzmann.

$$\boldsymbol{Pr\{E = E\} = \frac{1}{Z(T)} \cdot \exp\left(-\frac{E}{K_B T}\right)} \quad (8)$$

A medida que la temperatura baja la distribución de Boltzmann se concentra en los estados con menor energía y finalmente cuando la temperatura se acerca a cero, solo el estado con mínima energía tiene probabilidad diferente de cero. Si el proceso de enfriamiento no llega a ser lo suficientemente lento no se tendría un equilibrio térmico lo que podría llevar a la congelación del sólido y a formar una estructura amorfa meta-estable.

Para controlar y simular la evolución del equilibrio térmico se propone el Método de MonteCarlo el cual genera una secuencia de estados del sólido. Se propone que el proceso se realice por etapas donde se simula la temperatura, cada etapa del proceso consiste en cambiar la configuración hasta alcanzar un equilibrio térmico donde se inicializa una nueva etapa con una temperatura aun más baja y la solución del problema en general es la configuración obtenida en la última etapa.

En el método de Montecarlo, dado un estado actual *i* del sólido se elige una partícula de manera aleatoria a la cual se le aplica un pequeño desplazamiento produciendo una pequeña perturbación en la configuración del sólido, conocida como estado *j*. Como consecuencia de dicho cambio existe una diferencia de energía $\Delta E_{ij} = E_j - E_i$ entre el estado actual y la nueva configuración. Si la diferencia de energía es negativa, es decir E_j es menor que E_i, el proceso continuaría tomando como partida la nueva configuración, que se convertiría en el estado actual del sólido; en caso que $\Delta E_{ij} > 0$, se aplicaría el criterio de metrópolis, que considera una probabilidad de tomar como estado actual la nueva configuración dada por $\exp(-(\Delta E/KbT))$.

El algoritmo de SA puede ser definido como la secuencia de evaluación del criterio metrópolis para diferentes valores de temperatura que varían según un esquema de enfriamiento. Dicha evaluación permite que el algoritmo acepte cualquier solución al comienzo de la búsqueda, mientras que solo las buenas soluciones tendrán una mayor probabilidad de aceptación a medida que se itera.

Este algoritmo de SA no garantiza alcanzar una solución óptima, pero permite escapar de mínimos locales por la incorporación de una función de probabilidades en la aceptación o rechazo de nuevas soluciones aún cuando hay un cambio positivo de energía, así se garantiza que se llegue a una solución buena aunque no sea la mejor. Una de las grandes ventajas de este algoritmo es que no necesita una memoria de gran tamaño para ser ejecutado, y es usado para mejorar la convergencia de Algoritmos Geneticos (AG) mediante pruebas de los miembros de la población después de cada generación. Su ejecución es eficiente y fiable, debido a que se trata de una estrategia eficiente para elegir estados futuros aleatoriamente.

2.6. Máquina de Boltzmann

La máquina de Boltzmann es una red neuronal de tipo estocástico. Las diferentes arquitecturas posibles de este tipo de red pueden resumirse en tres tipos principales de modelos de la máquina Boltzmann, siendo estos las redes *completion network, input-output network,* y la *feedforward network.* La información consignada para la sección 2.6 fue traducción libre de [4].

Puede notarse como una red B, como un conjunto V neuronas y un conjunto C de conexiones entre las mismas. Los elementos (neuronas) son binarios, es decir, pueden tener dos valores de salida {0,1} llamados estados activo e inactivo respectivamente. Una configuración *k* de la máquina es un vector binario de longitud |V|, tal que el elemento *k(v)* representa el estado de la neurona *v*. Se define a su vez el conjunto *K*, que contiene a todas las configuraciones *k* posibles para B.

Cada conexión (v_1,v_2) de C tiene un peso simétrico asociado $w_{vivj}=w_{vjvi} \in R$. Dichas conexiones se consideran activadas cuando tanto *k(v1)* como *k(v2)* son igual a 1. Se define una función F (9) llamada función de consenso como una medida cuantitativa que mide la calidad de la configuración *k* de la red como:

$$\boldsymbol{F(k)} = \sum_{(\boldsymbol{v_1},\boldsymbol{v_2})\in \boldsymbol{C}} \boldsymbol{w_{v_1v_2}} \cdot \boldsymbol{k(v_1)} \cdot \boldsymbol{k(v_2)}$$

(9)

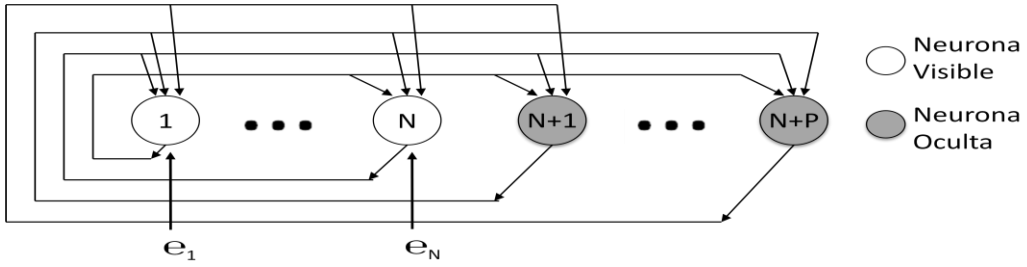
La máquina de Boltzmann tiene como propósito la maximización de la función de consenso, lo que podría interpretarse de su definición como una tendencia de las conexiones exitatorias (w_{vivj} positivo) a ser activadas, y de las conexiones inhibitorias (w_{vivj} negativo) a ser evitadas.

2.6.1. Arquitecturas comunes de la máquina de Boltzmann

Como se mencionó las máquinas de Boltzmann más encontradas en la literatura son las redes *completion network,input-output network,* y la *feedforward network.*

La red *completion network* tiene una arquitectura similar a la red Hopfield con una diferencia fundamental que es la posibilidad de encontrar neuronas tanto visibles como ocultas, es decir que no se trata de una arquitectura monocapa como propuso Hopfield. Las neuronas ocultas son aquellas no accesibles desde el exterior de la red, lo que implica que no reciben entradas externas al sistema ni son puntos de salida del mismo. Asimismo, las conexiones se establecen bidireccionalmente con pesos simétricos ($w_{vivj}=w_{vjvi}$).

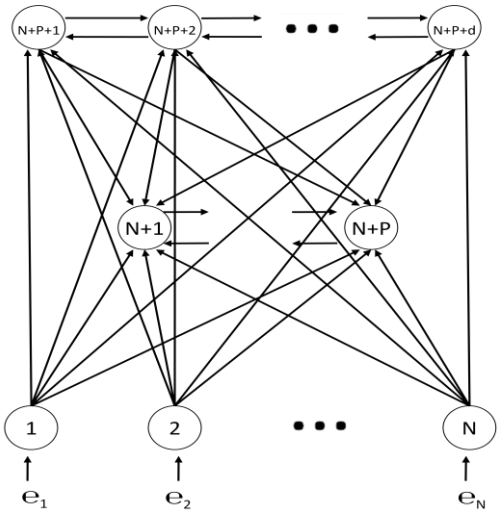
Figura 7. Completion Network



Fuente: Modificado de [1].

La red *input-output* al igual que la *completion network* tiene neuronas visibles y ocultas, las primeras en la capa de entrada y salida y las segundas en una capa intermedia. Este tipo de máquina de Boltzmann tiene tres tipos de conexiones. Las conexiones hacia adelante son aquellas entre la capa de entrada y la capa oculta, y entre la capa de entrada y la salida. Las conexiones en ambos sentidos son aquellas entre las neuronas ocultas y las de la capa de salida o de entrada. Finalmente, las conexiones laterales son aquellas entre neuronas de la misma capa.

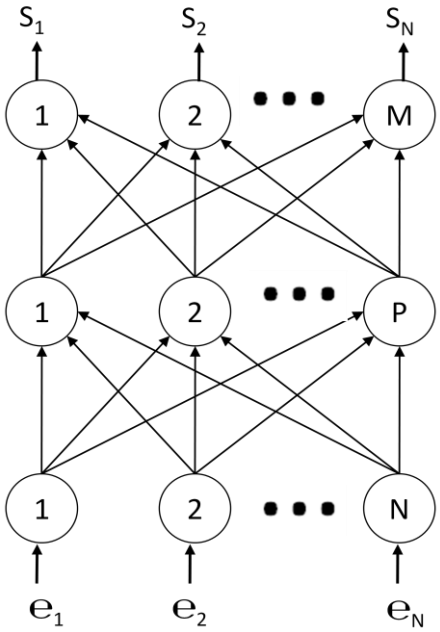
Figura 8. Input-Output Network



Fuente: Modificado de [1]

Por último, la *feedforward network* es una máquina de Boltzmann que cuente únicamente con conexiones hacia adelante, similar a un perceptrón de 3 capas.

Figura 9. Feed Forward Network



Fuente: Modificado de [1]

2.6.2. Mecanismos de transición

El mecanismo de transición gobierna el modo en que los elementos de la red cambian de estado en busca de un máximo en el consenso. Para introducir este concepto debe plantearse la diferencia entre máquinas de Boltzmann secuenciales y paralelas. En el primer caso, la salida de las neuronas cambia una a la vez, mientras que en el segundo los elementos de la red pueden cambiar su salida de forma simultánea.

Analizando el caso de una máquina secuencial se define la vecindad de cada configuración *k* como N_k, que representa el conjunto de configuraciones tales que cambia el estado de sólo un elemento respecto a *k.* La diferencia en el consenso entre dos configuraciones k_1 y k_2 debido a un cambio en el elemento v, asumiendo $k_1(v) = 1$ y $k_2(v) = 0$ sería

(10).

$$\Delta F_{k_1}(v) = [1 - 2 * k_1(v)]\left(\sum_{(v,u)\in C} w_{vu} k_1(u) + w_{vv}\right) \quad \textbf{(10)}$$

Si se asume $k_1(v)=0$ y $k_2(v)=1$, la expresión sería de la misma forma con signo negativo.

Se llama a *k* un mínimo local si $\Delta F_{k1} \leq 0$ para todo $v \in V$, es decir, que un cambio en un sólo elemento de la neurona no aumenta el consenso de la red.

El mecanismo de transición de estados se define entonces así: para una configuración k1 alcanzada se selecciona aleatoriamente un elemento v y se calcula la diferencia en el consenso causada por un cambio de estado en dicho elemento. Ahora, dependiendo del valor de ΔF_{k1} y un parámetro de control c>0 se decide si el cambio es aceptado. Este mecanismo se asemeja al algoritmo de *simulated annealig* expuesto anteriormente, definiendo una probabilidad de aceptación de la transición de k1 a k2, P, como se tiene en

(11):

$$\boldsymbol{P_{k_1}(v,c) = \frac{1}{1 + \exp\left(\frac{-\Delta F_{k_1}(v)}{c}\right)}} \quad (11)$$

Si la probabilidad de aceptación es mayor a un valor aleatorio entre cero y uno, se aceptará el cambio de estado propuesto. [5]

Se elige un valor inicial de c relativamente alto. El proceso se repite iterativamente disminuyendo lentamente el valor de c, determinando para cada valor un número fijo de iteraciones T. El proceso finaliza cuando se alcanza un número L de iteraciones consecutivas sin un cambio en el consenso. La forma en que disminuye c se denomina programa o itinerario de enfriamiento, y puede ser determinado antes de comenzar la iteración, así como cambiar dependiendo del comportamiento de la red. Un programa de enfriamiento demasiado brusco (T pequeño y cambio rápido de c) puede llevar a que la red se estanque en un mínimo local y la solución sea de baja calidad; sin embargo, un programa de enfriamiento demasiado suave (T grande y cambio lento de c) podría terminar en un tiempo de cómputo excesivo e innecesario.

Puede demostrarse matemáticamente que dado un tiempo de cómputo suficiente, la máquina de Boltzmann converge siempre al máximo global, y que en un tiempo menos a este siempre se llega a un mínimo local.

Se introduce para las redes simultáneas los conceptos de paralelismo sincrónico y asíncrono. En el primer caso se programan conjuntos de transición de estados propuestos por un subconjunto de neuronas, que se evalúan simultáneamente. Luego, los cambios aceptados se comunican a través de la máquina, lo cual implica que para el siguiente conjunto de transiciones se conoce la configuración exacta de la red. El sincronismo implica que debe contarse con un esquema global de reloj.

Para el caso asíncrono los elementos generan cambios de estado continuamente que son evaluados con base en información que no está necesariamente actualizada dado que el estado de elementos conectados a estos puede haber cambiado en el proceso, esto porque los conjuntos de cambios de estado en distintos subconjuntos de la red no se programan sucesivamente, ni se espera a que la información de transiciones sea propagada por toda la máquina de la red antes de dar cabida a cambios de estado en otros subconjuntos.

Se habla de paralelismo limitado cuando sólo es permitido que elementos no conectados cambien de estado de forma simultánea, y de paralelismo ilimitado cuando no se tiene dicha restricción.

Considérese una máquina de Boltzmann sincrónica con paralelismo limitado. En este caso al realizar particiones de los elementos, que corresponden a conjuntos del mayor tamaño posible en donde no haya dos elementos conectados, se obtiene la mayor velocidad de la red ya que todos los elementos pueden proponer transiciones de estado simultáneamente.

En redes en que se implementa paralelismo ilimitado se corre el riesgo de aceptar transiciones no deseadas debido a errores en el cálculo de ΔF_{k1}, aunque la probabilidad de aceptar transiciones indeseadas disminuye conforme disminuye c.

Puede demostrarse matemáticamente que si hay paralelismo limitado la máquina de Boltzmann converge asintóticamente a la configuración óptima, aseveración que no puede hacerse al presentarse paralelismo ilimitado.

2.6.3. Solución de problemas de optimización combinatoria usando la máquina de Boltzmann

Para resolver un problema de optimización combinatoria debe diseñarse la red de forma que esta represente apropiadamente el problema, el cual puede definirse en función de variables binarias. Asignando a cada elemento de la red dichas variables (xi), una configuración de la Máquina de Boltzmann define una solución no necesariamente factible del problema de optimización, siendo xi=k(vi).

La factibilidad de la función de consenso se alcanza cuando cada mínimo local de la máquina de Boltzmann corresponde a una solución factible del problema. De esta forma cuando la función de consenso es factible se garantiza que la red encuentra una solución factible al problema combinatorio.

Debe definirse, en aras de representar correctamente el problema en la máquina de Boltzmann, una función m(k), tal que dicha función permita "mapear" el conjunto de configuraciones K, en un conjunto de soluciones L del problema, definiendo a su vez como L' el conjunto de soluciones factibles del mismo. Otra función f(m(k)), llamada función de costo, asigna un número real a cada solución, permitiendo comparaciones cuantitativas entre las mismas.

La función de consenso debe escogerse de forma que la calidad de óptimos locales de la red refleje la calidad de las soluciones al problema de optimización, esto es, que preserve el orden. De manera más formal, se dice que la función de consenso preserva el orden, cuando para cualquier par de configuraciones k1 y k2 que pertenecen a K, se cumple

(12).

$$\boldsymbol{f(m(k1)) > f(m(k2)) \leftrightarrow F(k_1) > F(k_2)}$$

(12)

En caso de que representemos un problema de maximización. En caso contrario, de un problema de minimización, la preservación del orden estaría dada por

(13).

$$\boldsymbol{f(m(k1)) < f(m(k2)) \leftrightarrow F(k_1) > F(k_2)}$$

(13)

2.6.4. Implementación del problema del vendedor viajero usando la máquina de Boltzmann

Considérese el siguiente problema. Dadas n ciudades y una distancia c_{ij} entre cada par de ciudades (i,j). Determínese la mínima distancia posible de un tour que visite todas las ciudades, pasando como máximo una vez por ciudad.

Una posible formulación matemática binaria del problema podría ser

$sujeto\ a\ \sum_{i=1}^{n} x_{ip} = 1\ para\ todo\ p,$

(a)

$$\sum_{p=1}^{n} x_{ip} = 1\ para\ todo\ i,$$

(b)

$$x_{ip}\ \epsilon\ \{0,1\}\ para\ todo\ i,p$$

(c):

$$\min \sum_{i,j,p,q=1}^{n} d_{ijpq}\, x_{ip} x_{jq}$$

$$sujeto\ a\ \sum_{i=1}^{n} x_{ip} = 1\ para\ todo\ p,$$

(a)

$$\sum_{p=1}^{n} x_{ip} = 1\ para\ todo\ i,$$

(b)

$$x_{ip}\ \epsilon\ \{0,1\}\ para\ todo\ i,p$$

(c)

Con variables:

$$x_{ip} = \begin{cases} 1 & si\ el\ recorrido\ visita\ la\ ciudad\ i\ en\ la\ p-ésima\ posición \\ 0 & de\ otra\ forma \end{cases}$$

(d)

Y parámetros:

$$d_{ijpq} = \begin{cases} c_{ij}\ si\ q = mod((p+1), n) \\ 0\ \ de\ otra\ forma \end{cases}$$

(e)

(14)

De esta forma, se tiene que para cada recorrido por las ciudades determinado por las variables x_{ip}, se sabe que $d_{ijpq}\, x_{ip} x_{jq} = c_{ij}$ sí y solo sí la ciudad *i* precede inmediatamente a la ciudad *j* en el recorrido. De otra forma $d_{ijpq} = 0$. La distancia total recorrida es $\sum_{i,j,p,q} d_{ijpq}\, x_{ip} x_{jq}$.

Para que la máquina de Boltzmann pueda resolver el problema del vendedor viajero se necesita un elemento v_{ip} por cada variable x_{ip}. Adicionalmente se define un conjunto de conexiones $C_1 \cup C_2 \cup C_3$ de la siguiente forma (15):

$$C_1 = \{(v_{ip}, v_{ip})\}\ (bucles) \quad (a)$$

$$C_2 = \{(v_{ip}, v_{jq}) | i \neq j\ y\ q = mod((p+1), n)\} \quad (b)$$

$$C_3 = \{(v_{ip}, v_{jq}) | (i = j\ y\ p \neq q) ó\ (i \neq j\ y\ p = q)\} \quad (c)$$

Si se escogen los pesos de las conexiones de la siguiente manera:

Para todo $(v_{ip}, v_{ip}) \in C_1, se\ toma\ w_{v_{ip}\,v_{ip}} > max_{k,l,k\neq l}(c_{ik} + c_{il})$ para evitar bucles.

Para todo $(v_{ip}, v_{jq}) \in C_2, se\ toma\ w_{v_{ip}\,v_{jq}} = -c_{ij}$.

Para todo $(v_{ip}, v_{jq}) \in C_3, se\ toma\ w_{v_{ip}\,v_{jq}} < -\min\left\{w_{v_{ip}\,v_{ip}}, w_{v_{jq}\,v_{jq}}\right\}$.

3. ESPECIFICACIONES

3.1. Algoritmo para la solución de CAP basado en redes de Hopfield (Hopfield-Funabiki)

3.1.1. Entradas y salidas del algoritmo

Los parámetros de entrada para Hopfield-Funabiki son:

Matriz de compatibilidad Mcomp. Esta matriz representa las restricciones de compatibilidad electromagnética, y es de dimensiones *nxn*,siendo *n* el número de celdas del escenario a solucionar.

Vector de demanda Vdem. Representa la demanda asociada a cada celda del escenario.

Número de canales *m.* Cantidad de canales disponibles para la asignación.

Se toma como salida la matriz V que almacena los estados de todas las neuronas y representa la respuesta al CAP. Esta representación de entradas y salidas se ha tomado de la representación neuronal del CAP (Sección 4.1.1)

3.1.2. Problemas a solucionar

Se propone para este algoritmo calcular la frecuencia de convergencia para los escenarios enunciados en la sección 2.4.2. Las Instancias de Philadelphia son problemas puramente teóricos, mientras que el problema de Finlandia modela una red real implementada en Helsinki [8].

3.2. Simulated Annealing

El proceso de SA está inspirado en el proceso de recocido de los metales, el cual consiste en enfriar lentamente un sólido fundido por altas temperaturas hasta encontrar un estado de baja energía para un conjunto de partículas en el sólido; para el caso de este algoritmo SA consiste en la generación aleatoria de soluciones al CAP comparando su energía a medida que la temperatura del sistema disminuye.

En términos generales se solucionará el CAP mediante del algoritmo de Hopfield-Funabiki iterativamente anidado en una rutina de simulated annealing, variando en cada repetición parámetros a los que se les asigna valores aleatorios de forma aleatoria. La energía de la solución se calcula con base en una función objetivo, buscando llegar siempre a una solución con la mínima energía posible. Soluciones consecutivas al CAP se comparan, aceptándose siempre como nueva respuesta las soluciones con menor energía, mientras que soluciones con mayor energía que la menor conocida, es decir, soluciones de menor calidad, son aceptadas probabilísticamente como método para escapar de mínimos locales.

La función objetivo es la solución de CAP con el algoritmo de Hopfield-Funabiki, y la forma de optimización la implementamos con tres métodos diferentes. El primero fue haciendo la comparación por

medio del número de violaciones que podría darnos cada caso, para este caso se busca minimizar la respuesta, nuestro objetivo será cero. El segundo fue enfocado en la convergencia, es claro que para este caso se busca maximizar la respuesta, hasta que llegue a uno. Y como tercero y último, para disminuir el tiempo de cómputo buscamos encontrar aquellos valores con los que la solución sea hallada en el menos número de iteraciones posible.

Dentro del proceso de Simulated Annealing la variación aleatoria de ciertos parámetros de entrada a la función objetivo es lo que nos permite tener soluciones diferentes para así compararlas y poder implementar el algoritmo. Tal como fue mencionado en la sección 5.1.2 al trabajar Hopfield-Funabiki, se descubrió que la variación de los coeficientes A, B y PesoC podrían llegar a generar mejoras sustanciales en la frecuencia de convergencia, por esta razón se escogieron estos parámetros para hacer la variación aleatoria de los parámetros de entrada de nuestra función objetivo.

3.2.1. Variables del algoritmo

Inicialización: Tmin, To, T_actual, épsilon, k y objetivo. Sea cual sea el valor de las variables en el momento que se encuentre la solución objetivo terminara la simulación, la cual arrojara parámetros de entrada óptimos para la función. Tmin será la temperatura mínima a la que podrá llegar para buscar solución, si encuentra la solución objetivo antes de llegar a esta se daría por terminado el proceso. To será la temperatura inicial de la simulación. T_actual, será la temperatura que tiene en el momento, la cual varía dependiendo de k, constante que incrementa cada vez que termina las épsilon iteraciones que realiza por temperatura al no encontrar solución objetivo. El objetivo lo consideramos como una variable para diferenciar el caso que se está realizando, ya sea de maximización donde objetivo será igual a uno (1), o de minimización donde su valor será cero (0).

Coeficientes A, B y PesoC: Entradas de la función objetivo, se escoge un valor entre 0 y 1 de forma aleatoria, que nos indicara los pesos de la ecuación de movimiento. También son actualizados ya que no solo nos conviene la solución, sino los parámetros con los que se llego a ella.

Generales: n, m, resultado, U, Umin, V. n será el número de celdas que es determinado para cada escenario, por medio de la lectura de los archivos .txt de cada una de las instancias se puede hallar fácilmente. m número de canales que serán asignados, para la solución a nuestro de asignación de canales se busca que sea la menor cantidad posible. resultado es un vector donde se guardan las violaciones de la matriz V, resultado=[demanda,CoSite,AdjacentChannel]. Umin es el límite inferior de saturación para la entrada. U, matriz de entrada. V, matriz que muestra el estado de cada una de las neuronas lo cual es la solución a nuestro problema.

3.3. Máquina de Boltzmann

Se propone implementar una máquina de Boltzmann secuencial. La razón para esta escogencia es que para este tipo de redes puede garantizarse que su configuración converge asintóticamente a un mínimo global dado para un tiempo lo suficientemente grande, a su vez, puede asegurarse que en un tiempo menor la red converge a una configuración de mínimo local [4].

3.3.1. Análisis propuestos

Se han propuesto dos análisis para la máquina de Boltzmann en el escenario F1. Estos análisis son:

Cambio de la frecuencia de convergencia en función de la temperatura inicial en ausencia de ruido aleatorio.

Incidencia de la cantidad de ruido introducido a la red y el porcentaje de neuronas actualizadas por valor de temperatura.

Se realizarán además simulaciones para las instancias de Philadelphia y Finlandia.

4. DESARROLLO

4.1. Algoritmo de solución de CAP basado en redes de Hopfield (Funabiki)

Funabiki propuso en [7] un algoritmo para la solución de CAP en una red de comunicaciones de *n* celdas contando con un número *m* de canales. Esta asignación como se ha expuesto está sujeta a un requerimiento de demanda por cada celda así como a un conjunto de restricciones de compatibilidad electromagnética.

En este planteamiento la demanda o número de canales asignados requeridos para cada celda es representada por un vector Vdem, siendo Vdem(*i*) la demanda para la celda *i*. Una matriz Mcomp, representa tres tipos de restricciones de compatibilidad electromagnética, siendo estas:

Restricción *co-channel:* establece que el mismo canal no puede ser asignado a ciertos pares de celdas simultáneamente.

Restricción *adjacent-channel:* establece que canales adyacentes en frecuencia no pueden ser asignados simultáneamente a ciertos pares de celdas de forma simultánea.

Restricción *co-site:* establece que todo par de canales asignados a una misma celda deben estar separados en frecuencia por una distancia mínima determinada.

Mcomp tiene dimensiones de *nxn*, siendo Mcomp(*i,j)* el valor de separación entre un canal asignado a la celda *i*, y otro asignado al a celda *j*.

4.1.1. Representación neuronal de CAP

El CAP para el sistema propuesto se modela mediante una red neuronal de *nxm* neuronas, siendo *n* y *m* el número de celdas y canales respectivamente planteadas para el escenario (*benchmark*) a solucionar. Una matriz *V* de dimensiones *nxm* representa los estados para todas las unidades de la red, es decir la solución al CAP. Cada elemento *V*(*i*,*j*) de esta matriz toma el valor de 1 si el canal *j* se asignó a la celda *i* y 0 en caso contrario. Una matriz *U* de dimensiones *nxm* representa las entradas a las neuronas de la red que determinarán el estado de las mismas con base en una función de activación. *U*(*i*,*j*) es la entrada para la neurona (*i, j).*

Para satisfacer la restricción de demanda (16) debe garantizarse que para cada celda *i* hayan *Vdem*(*i*) neuronas activadas, es decir con salida igual a 1.

$$\sum_{q=1}^{m} V(i,q) - V_{dem}(i) = 0, para\ i = 1,2,..,n \quad \textbf{(16)}$$

La restricción *co-site*

(17) establece que si el canal q ha sido asignado a la celda i, cualquier otro canal asignado a la misma celda debe estar separado por una distancia de al menos Mcomp(i,i) del canal q.

$$\sum_{\substack{q=j-(\boldsymbol{Mcomp}(i,i)-1) \\ q\neq j \\ 1\leq q\leq m}}^{j+(\boldsymbol{Mcomp}(i,i)-1)} \boldsymbol{V(i,q)=0, para\ i=1,2,\dots,n}$$

(17)

Las restricciones *co-channel* y *adjacent-channel* (18) implican que si un canal q ha sido asignado a la celda i, cualquier canal asignado a la celda p debe estar separado de q por una distancia de al menos Mcomp(i,p).

$$\sum_{\substack{p=1 \\ p\neq i \\ Mcomp\ (i,p)>0}}^{n} \sum_{q=j-(Mcomp\ (i,p)-1)}^{j+(Mcomp\ (i,p)-1)} V(p,q)=0, para\ i=1,2,,\dots,n$$

(18)

El cambio en las entradas U de las neuronas se determina por la incidencia del estado de las neuronas en la energía de la red de la energía de la red, este cambio en la energía se conoce como ecuación de movimiento notada como DeltaU (19). El valor DeltaU(i,j) representa el cambio calculado para la entrada de la unidad (i,j).

$$DeltaU(i,j)=-A*f\left(\sum_{q=1}^{m}V(i,q)-V_{dem}(i)\right)-B\left(\sum_{\substack{q=j-(Mcomp\ (i,i)-1) \\ q\neq j \\ 1\leq q\leq n}}^{j+(Mcomp\ (i,i)-1)} V(i,q)+\sum_{\substack{p=1 \\ p\neq i \\ Mcomp\ (i,p)>0}}^{n}\ \sum_{q=j-(Mcomp\ (i,p)-1)}^{j+(Mcomp\ (i,p)-1)} V(p,q)\right)$$

$$+C*h\left(\sum_{q=1}^{m}V(i,q)-V_{dem}(i)\right)*\big(1-V(i,j)\big)$$

(19)

El primer término de la ecuación de movimiento es ponderado por A y asegura que para cada celda i un número Vdem(i) de elementos tengan salida diferente de cero. El segundo término es ponderado por B y evita que la ij-ésima neurona tenga salida diferente de cero si la asignación del canal j a la celda i viola alguna de las restricciones de compatibilidad mencionadas anteriormente. Dicho de otra forma los primeros dos términos de la ecuación de movimiento representan completamente las restricciones de demanda y compatibilidad electromagnéticas impuestas al CAP. El tercer y último término de la ecuación de movimiento es ponderado por C y tiene el fin de aumentar la probabilidad de que la ij-ésima neurona tenga salida diferente de cero si la celda i tiene menos de Vdem(i) canales asignados.

A,B y C son constantes que ponderan los aportes de las distintas restricciones a la ecuación de movimiento. *f(x)* y *h(x)* son funciones conocidas como heurísticas de convergencia [7].

La salida de la red V se actualiza con una función de activación dada por el modelo de McCulloch-Pitts (20) que establece que,

$$V(i,j) = \begin{cases} 1 \; si \; U(i,j) > UTP(upper \; tree \; point) \\ 0 \; si \; U(i,j) < LTP(lower \; tree \; point) \\ no \; cambia \; de \; otra \; forma \end{cases}$$

(20)

UTP y LTP son los umbrales de activación y desactivación de las neuronas respectivamente.

Se presenta a continuación la rutina general para la implementación de una red neuronal tipo Hopfield para la solución de CAP.

Paso 1. Hacer t=1, A=B=1, T=10, w=5, Umax=30, Umin=-30, UTP=5, UTP=-5, Tmax=500.

Paso 2. Iniciar las entradas de la red U con valores aleatorios entre 0 y Umin, y las salidas V con valores igual a 0.

Paso 3. Asignar C=3,4,o 5 de forma aleatoria y calcular la ecuación de movimiento para cada neurona de acuerdo a (21).

para $mod(t,T) < w$

$$DeltaU(i,j) = -A * f\left(\sum_{q=1}^{m} V(i,q) - V_{dem}(i)\right) - B\left(\sum_{\substack{q=j-(Mcomp\,(i,i)-1) \\ q\neq j \\ 1\leq q\leq m}}^{j+(Mcomp\,(i,i)-1)} V(i,q) + \sum_{\substack{p=1 \\ p\neq i \\ Mcomp\,(i,p)>0}}^{n} \sum_{q=j-(Mcomp\,(i,p)-1)}^{j+(Mcomp\,(i,p)-1)} V(p,q)\right) * V(i,j)$$

$+C * h\left(\sum_{q=1}^{m} V(i,q) - V_{dem}(i)\right) * \left(1 - V(i,j)\right)$ **(a)**

para $mod(t,T) < w$

$$DeltaU(i,j) = -A * f\left(\sum_{q=1}^{m} V(i,q) - V_{dem}(i)\right) - B\left(\sum_{\substack{q=j-(Mcomp\,(i,i)-1) \\ q\neq j \\ 1\leq q\leq m}}^{j+(Mcomp\,(i,i)-1)} V(i,q) + \sum_{\substack{p=1 \\ p\neq i \\ Mcomp\,(i,p)>0}}^{n} \sum_{q=j-(Mcomp\,(i,p)-1)}^{j+(Mcomp\,(i,p)-1)} V(p,q)\right)$$

$$+C * h\left(\sum_{q=1}^{m} V(i,q) - V_{dem}(i)\right) * \left(1 - V(i,j)\right)$$

(21)

Paso 4. Actualizar las entradas U de las neuronas según el modelo de Euler (22).

$$\boldsymbol{U = U + DeltaU}$$

(22)

Paso 5. Saturar la nueva entrada, provista por (22), entre Umin y Umax. Funabiki denomina a este procedimiento heurística de saturación de la entrada, y tiene como propósito acelerar la convergencia de la red.

Paso 6. Actualizar los valores V de salida de la red según el modelo de McCulloch-Pitts (20).

Paso 7. Revisar si se cumplen las restricciones electromagnéticas para todas las neuronas y la condición de demanda o sí se ha llegado al número máximo de iteraciones Tmax, si es así, terminar el algoritmo, dando a *V* como solución, en caso contrario aumentar t en 1 y volver a 3.

Los valores A,B y C son los pesos de los términos de la ecuación de movimiento, T y w constantes que determinan la relación en que se utilizan los dos términos de la ecuación de movimiento, Umax y Umin

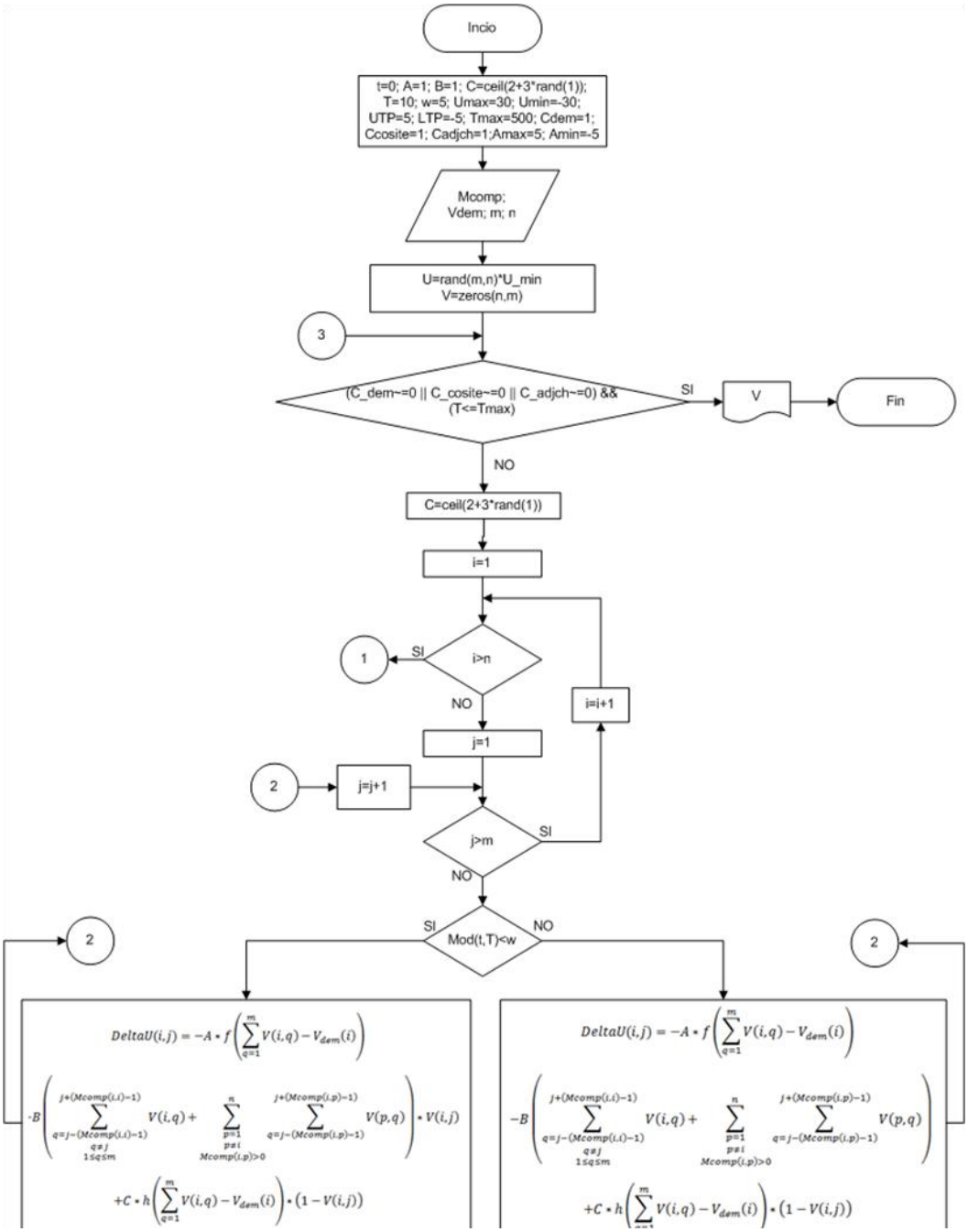
los límites de saturación de la entrada de las neuronas, t la iteración actual y Tmax el número máximo de iteraciones permitidas.

En el paso 3 pueden observarse dos formas de (21). Durante la mitad de las iteraciones se usa la forma original, y durante la otra mitad el B se multiplica por el estado de la neurona para la que se calcula la ecuación de movimiento. Esta estrategia pretende ayudar a escapar de mínimos locales, permitiendo ignorar las violaciones de interferencia co-channel y adjacent-channel para neuronas con salida cero cuando se aplica la primera forma de (21).

El paso 5 es llamado por Funabiki como heurística de saturación de la entrada, y tiene como propósito acelerar la convergencia de la red.

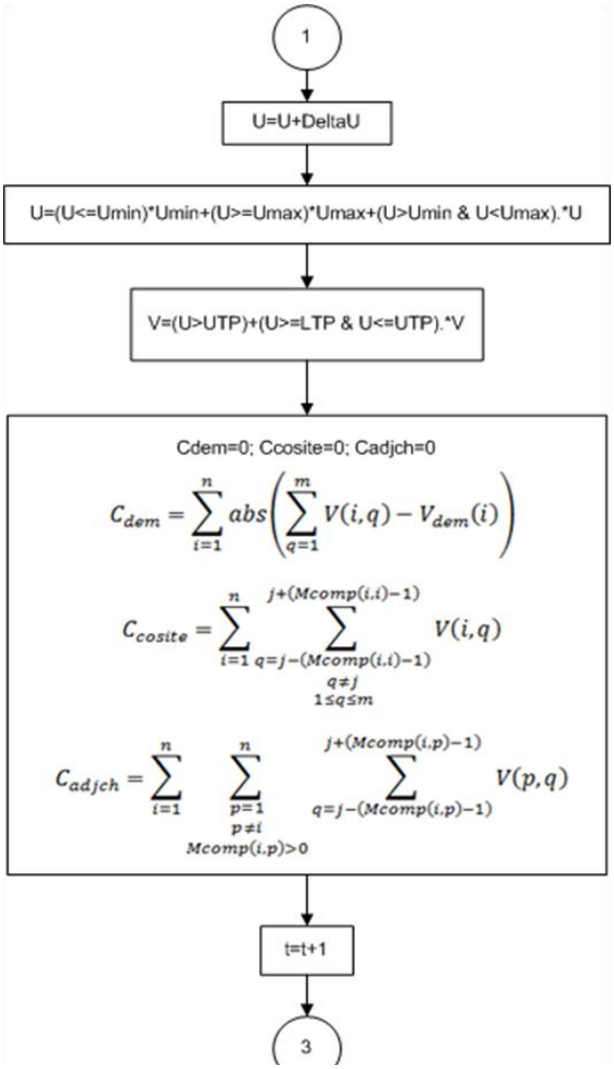
4.1.2. Diagrama de flujo

Diagrama 1. Funabiki (Primera parte)



Fuente: Presentación propia de los autores.

Diagrama 2. Funabiki (Segunda parte)



Fuente: Presentación propia de los autores.

4.2. Simulated Annealing

El algoritmo de SA está inspirado en el proceso de recocido de los metales, el cual consiste en enfriar lentamente un sólido fundido por altas temperaturas hasta encontrar un estado de baja energía para un conjunto de partículas en el sólido. [13]

En términos generales se solucionará el CAP iterativamente, variando cada vez parámetros asociados al mismo de forma aleatoria. Se comparan las soluciones provistas empleando estos parámetros aceptándose siempre como nueva respuesta al problema las soluciones con menor energía, mientras que soluciones con mayor energía que la menor conocida, es decir, soluciones de menor calidad, son aceptadas probabilísticamente como método para escapar de mínimos locales.

La energía de las soluciones se calcula según la función objetivo del CAP, buscando llegar siempre a una solución con la mínima energía posible. La función objetivo a optimizar será la solución al CAP provista por el algoritmo de Hopfield-Funabiki enfocándonos en el número de violaciones a las restricciones de demanda y compatibilidad electromagnética. De esta manera busca asegurarse que Hopfield-Funabiki llegue a una solución válida del CAP.

Los parámetros a variar aleatoriamente para producir soluciones al CAP mediante Hopfield-Funabiki serán las variables A,B y PesoC de la ecuación de movimiento (21), cuya variación tiene una fuerte incidencia en la frecuencia de convergencia del mismo.

Se enuncia a continuación el procedimiento para la solución del CAP mediante la rutina de SA-Hopfield:

Paso 1. Inicialización de variables T_min; To; T_actual=To; epsilon; k=0; objetivo.

Paso 2. Asignación de valores aleatorios a A,B PesoC. Evaluación de la función objetivo con dichos valores.

Paso 3. Terminar proceso si se ha llegado a una solución al CAP sin violaciones o la temperatura ha llegado a su mínimo valor. De ser válida la solución se almacenan los parámetros utilizados A,B,PesoC dando a V como solución.

Mientras T_actual>Tmin && i<epsilon

Paso 4. Nueva asignación de valores aleatorios y Evaluación de la función objetivo con dichos valores.

Paso5. Verificar el número de violaciones en la nueva solución al CAP. De ser cero se termina el proceso dando la nueva solución y los nuevos parámetros como respuesta al CAP. Si no se ha solucionado el problema, seguir a ***Paso6.***

Paso6. Comparación de las soluciones (Objetivo).

Si la nueva solución propuesta no es de mejor calidad que la mejor conocida, calcular

$$exp\left(-\frac{(Objetivo_{mut} - Objetivo)}{T\_actual}\right) > rand(0,1) \quad \textbf{(23)}$$

Paso7. Iteraciones.

Si los pasos 1 a 6 se han realizado épsilon veces, actualizar el valor de temperatura actual según (24)

$$T_{actual} = \frac{To}{k+1} \quad \textbf{(24)}$$

Aumentar k en 1 y volver a ***Paso2.***

De otra forma, aumentar épsilon en 1 e ir a ***Paso4.***

En el ***Paso1*** T_min es la temperatura mínima a la que llegará el proceso de annealing, siendo To la temperatura inicial; T_actual es la temperatura actual del proceso. De este valor de temperatura depende la variación de la probabilidad de aceptación de soluciones de menor calidad a la mejor conocida, siendo esta probabilidad mayor cuanta más alta la temperatur; epsilon representa el número de veces que se correrá el algoritmo de Hopfield-Funabiki por cada valor de temperatura; k es una constante con base en la cual se actualiza el valor de T_actual y objetivo representa para nuestro algoritmo la satisfacción de las restricciones al CAP, al tener cero violaciones se ha encontrado la solución.

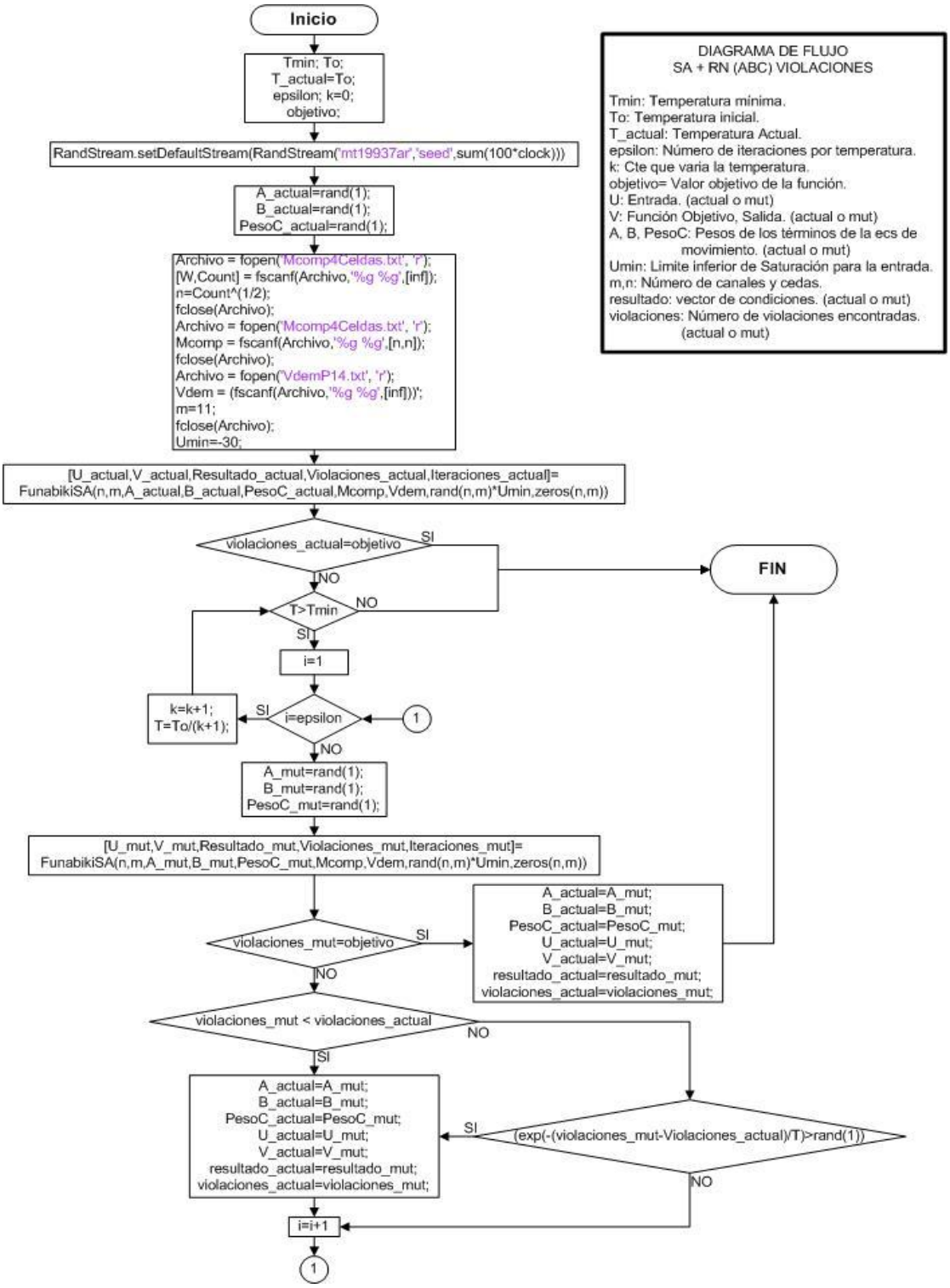
Los parámetros de entrada que se eligieron para variar de forma aleatoria son A, B y PesoC. En el desarrollo de Hopfield-Funabiki se presentó cómo la variación de estos parámetros es un aspecto fundamental de la obtención de altas frecuencias de convergencia. La variación se acotó entre 0 y 1.

En el ***Paso4*** se generan valores de A,B,PesoC y solución al CAP distintos a los mejores conocidos hasta el momento. Estos valores deben almacenarse en variables distintas a los mejores conocidos, marcados con la notación Actual. De esta forma si la nueva solución propuesta no es mejor que la mejor conocida, y no se acepta probabilísticamente, se sigue dando como respuesta al problema la mejor conocida.

Si se cumple la condición probabilística del ***Paso6*** se aceptará la nueva solución propuesta como la mejor conocida hasta el momento, aún cuando sea de menor calidad que la mejor solución conocida, y se almacenará en las variables originales que preservan la mejor propuesta. Por medio de este método se disminuye la probabilidad de caer en un estado de mínimo local.

4.2.1. Diagrama de flujo

Diagrama 3. Simulated Annealing



Fuente: Presentación propia de los autores.

4.3. Máquina de Boltzmann

El problema del viajero (TSP) expuesto en la sección 2.6.4tiene una gran similitud al CAP, siendo ambos problemas NP-difíciles. La arquitectura propuesta por [17] para el TSP sirvió como punto de partida para el desarrollo de la arquitectura de la red para la solución del CAP. La matriz *Mcomp* y el vector *Vdem,* así como las restricciones para CAP tratadas en la representación neuronal del CAPdefinen completamente el problema de optimización a solucionar.

4.3.1. Representación Neuronal para MB

La máquina de Boltzmann es una red neuronal de tipo estocástico. La representación neuronal del CAP para esta basada en dos funciones, una función de consenso que busca optimizarse y una función de activación o transición probabilística. Para un CAP de *n* celdas y *m* canales una matriz *Estados* de dimensiones *nxm* representa los estados o configuración de las *nxm* unidades de la red.

Los problemas de optimización que se solucionan con la máquina de Boltzmann deben representarse en variables binarias, por tal razón cada una de las *nxm* neuronas de *Estados* pueden tomar valores de 1 o 0. Considérese la El valor de salida (estado, o *Estados(i,j))* de esta unidad es 1 si el canal *j* sea asignado a la celda *i* y 0 en caso contrario.

Cada par de neuronas de la máquina de Boltzmann tiene un peso o fuerza asociado a su conexión que determina qué tan deseable es que dicho par de unidades estén activas al mismo tiempo como se indica en las figuras 10 y 11. Un peso positivo indica que es conveniente que las dos neuronas tengan una salida de valor 1, estableciendo una conexión excitatoria. Si dicho escenario busca evitarse la fuerza de conexión será negativa y la conexión se conocerá cono inhibitoria.

4.3.2. Función de Consenso

El consenso (25) es una función que evalúa la calidad de la configuración de estados de la red de forma análoga a la función de energía. Sin embargo esta función debe ser maximizada, ya que un consenso alto indica que conexiones exitatorias están activas, mientras conexiones inhibitorias no lo están.

$$C(Estados) = \sum_{i,j,p,q} S_{ij,pq}\, Estados(i,j) Estados(p,q) \quad \textbf{(25)}$$

Siendo $S_{ij,pq}$ el peso de la conexión entre las unidades *(i,j)* y *(p,q).*

Se define de manera similar el cambio en el consenso **(26)** para cada neurona de la red. Esta medida indica qué tanto cambia el consenso de la red al cambiar el estado de una unidad.

$$\Delta C(i,j) = [1 - 2Estados(i,j)] * [\sum_{p,q} S_{ij,pq}\, Estados(p,q) + S_{ij,ij}] \quad \textbf{(26)}$$

4.3.3. Probabilidad de Transición

A diferencia del modelo utilizado en HOPFIELD-FUNABIKI y SA-HOPFIELD en MB la función de activación no determina si una neurona debe o no estar activa, sino indica si un cambio en su estado. Como se indicó esta función de activación o de transición (28) es de tipo estocástico, por lo que está asociada a una distribución probabilística (27) conocida como función de Fermi [5] que depende del cambio en el consenso para la unidad analizada y un parámetro de temperatura.

$$P_{act} = \frac{1}{1 + e^{-\Delta C(i,j)/T}}$$

(27)

$$Estados(i,j) = \begin{cases} \neg Estados(i,j) & si\ P_t > rand[0,1] \\ no\ cambia\ en\ otro\ caso \end{cases}$$

(28)

T es un parámetro de temperatura que cambia cuando la red se ha actualizado total o parcialmente, de manera muy similar al proceso de SA donde un valor alto de temperatura implica una alta probabilidad de aceptación de transiciones.

4.3.4. Conexiones entre Neuronas

Para cada par de neuronas $Estados_{ij}$ y $Estados_{pq}$ existen tres conjuntos de fuerzas asociadas a su conexión, $S_{co\_site} \cup S_{adj\_ch} \cup S_{Bias}$, dadas por (29) e ilustradas en las gráficas 12 y 13.

$$S_{co\_site} = -M_{comp}(i,i) * \left[M_{comp}(i,i) - |j-q|\right]\ si\ i = p$$

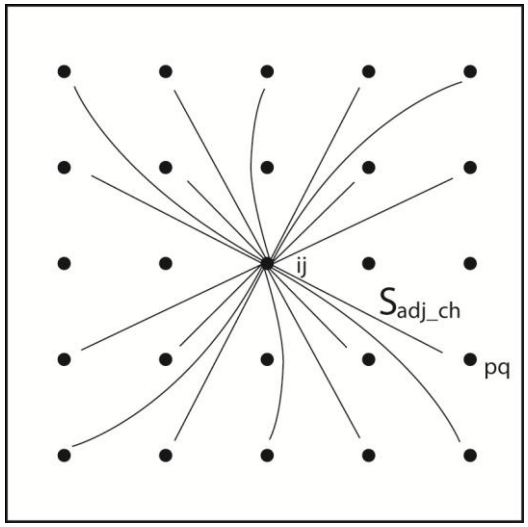
$$S_{adj\_ch} = -M_{comp}(i,p) * \left[M_{comp}(i,p) - |j-q|\right] si\ i \neq p \wedge (M_{comp}(i,p) > 0)$$

$$S_{Bias} = V_{dem}(i) * h(\sum_{j=1}^{n} Estados(i,j) - V_{dem}(i))\ para\ i = p\ \wedge j = q$$

$$h(x) = \begin{cases} 1\ para\ x < 0 \\ -1\ para\ x \geq 0 \end{cases}$$

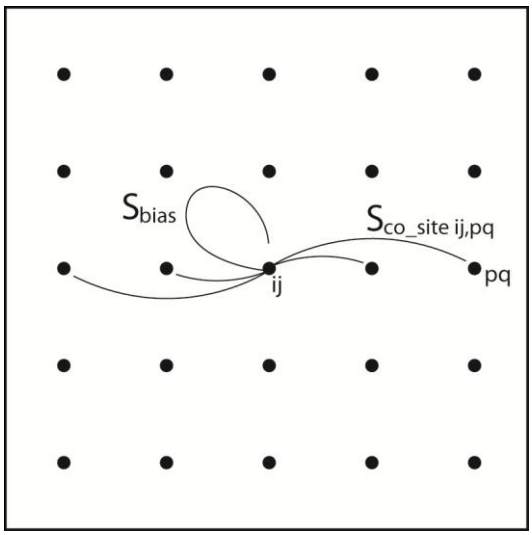
(29)

Figura 14: Esquema de conexiones adjacent channel



Fuente: presentación propia de los autores

Figura 15: Esquema de conexiones co-site y Bias



Fuente: presentación propia de los autores

Estas fuerzas representan las restricciones de demanda y compatibilidad electromagnética de CAP en la función de consenso. A diferencia de la ecuación de movimiento (19), sólo la fuerza S_{Bias} puede producir un cambio positivo en el consenso de la red, razón por la cual se ha asociado su valor al cumplimiento de la restricción de demanda. Las fuerzas S_{co_site} y S_{adj_ch} se han asociado a las restricciones de compatibilidad electromagnética y tienden a inhibir la activación de conexiones que introduzcan violaciones a la red.

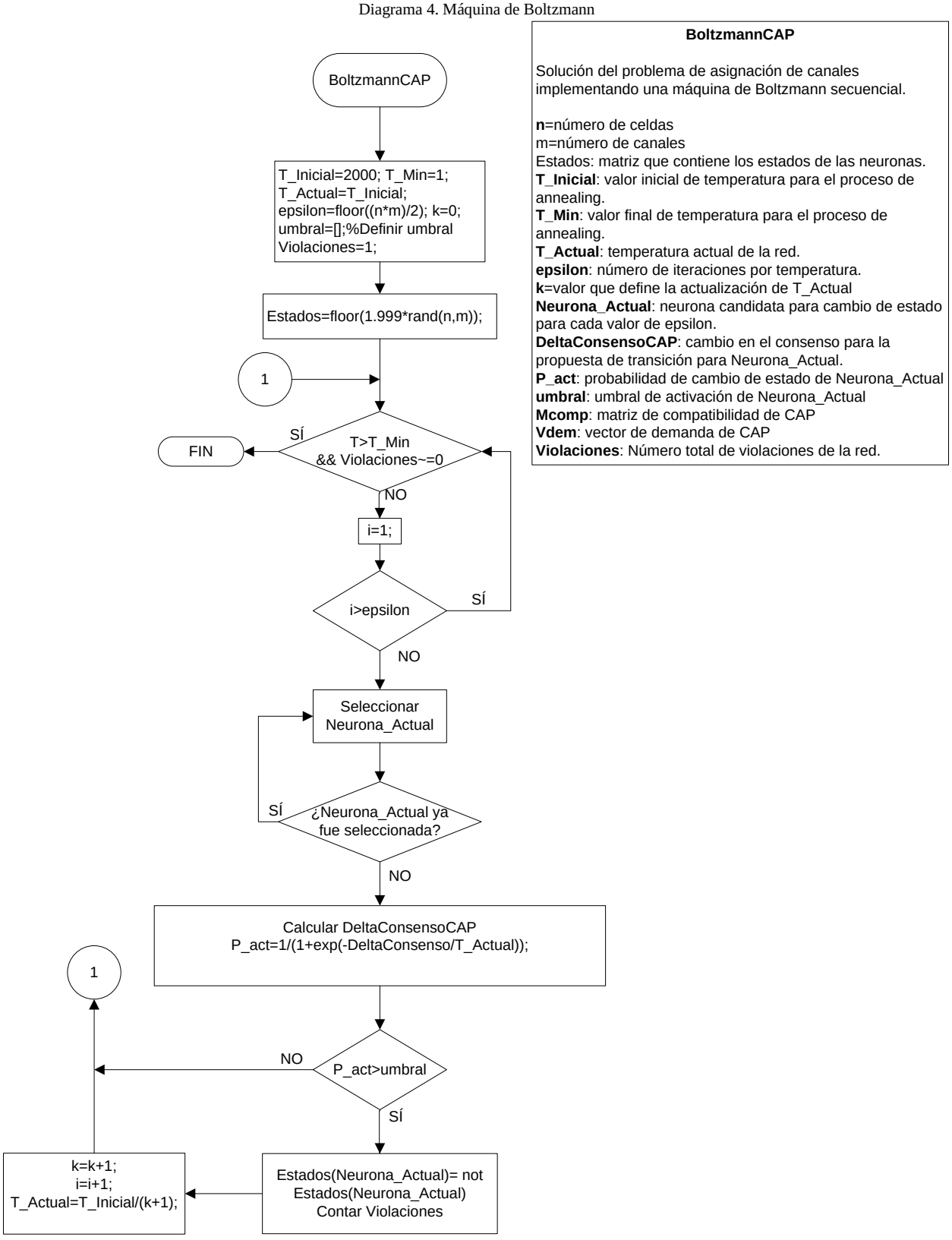
4.3.5. Porcentaje de Actualización y Ruido introducido en la Red

En el proceso de annealing anidado en la máquina de Boltzmann se ha propuesto que para cada valor de temperatura de la red se actualice un porcentaje de unidades de la misma en orden aleatorio.

Como mecanismo para escapar de mínimos locales se propuso la introducción de ruido a la red de forma aleatoria. Cuando la temperatura de la red ha decaído a un porcentaje fijo de la temperatura inicial, se selecciona de forma aleatoria un porcentaje de las unidades de la red, cambiando su estado sin contemplar el cambio en el consenso producido por este cambio.

4.3.6. Diagrama de flujo

Diagrama 4. Máquina de Boltzmann



Fuente: Presentación propia de los autores

4.3.7. Procedimiento general para la solución del CAP por medio de la máquina de Boltzmann (MB)

Paso 1. Definición de constantes de inicio. T_Inicial=1000; T_Min=1; k=0; epsilon=n*m;.

Paso 2.Inicialización aleatoria de la matriz Estados. Estados(i,j)={0,1} para todo i,j. Se inicia la máquina de Boltzmann en una configuración aleatoria.

Paso 3. Seleccionar una neurona $Estados_{ij}$ de manera aleatoria. Para cada valor de T_Actual, epsilon neuronas son elegidas secuencialmente de forma aleatoria.

Paso 4 .Calcular el cambio en el consenso DeltaConsenso al cambiar de estado la neurona $Estados_{ij}$.

Paso 5.Calcular la probabilidad de activación P_act con base en DeltaConsenso para $Estados_{ij}$.

Paso 6. Actualizar probabilísticamente la salida de $Estados_{ij}$. $Estados(i,j) = not\ Estados(i,j)\ \ si\ \ P_{act} > rand[0,1]$

Paso 7. Verificar si persisten violaciones en la red. Si no existen violaciones, se termina el procedimiento, si existen,

Paso 8. Si no se han seleccionado n*m neuronas, ir a 3.
si se han seleccionado ya n*m neuronas, actualizar el valor de T_Actual.
k=k+1; T_Actual=T_Inicial/(k+1)

Paso 9.Si T_Actual>T_Min, ir a 3. En caso contrario, terminar.

En el ***Paso1*** T_Actual, T_inicial y T_min son las temperaturas actual, inicial y final respectivamente entre las que la red iterará hasta encontrar una solución válida, epsilon es la cantidad de neuronas actualizadas por valor cada valor de temperatura de la red T_actual, k es una constante que aumenta cada vez que se han actualizado epsilon neuronas y actualiza el valor de temperatura

La probabilidad de activación del ***Paso5*** depende del valor de T_Actual de manera simular que en el algoritmo de SA. Para valores altos de temperatura P_act es más alta para valores menos favorables de cambio en el consenso. A medida que T_Actual disminuye, es menos probable que se acepte un cambio en el estado de una neurona si dicho cambio no aporta un cambio favorable en el consenso.

5. ANALISIS DE RESULTADOS

5.1. Hopfield-Funabiki

Se simuló el algoritmo de Hopfield-Funabiki sin ninguna modificación respecto a [7] para las 19 instancias de Philadelphia (P1 a P19) y la instancia de Finlandia (P20). La frecuencia de convergencia, es decir el porcentaje de casos en que la red neuronal convergió a una solución válida en cien simulaciones para el número de canales propuesto por dicho autor fue 0%. No se llegó a los resultados esperados. Al ver que los reportes de convergencia publicados para este algoritmo eran en algunos casos de 100% se concluyó que existen tres factores principales a tener en cuenta al implementar la red. Estos factores son los pesos de los términos de la ecuación de movimiento (19), la posibilidad de asignar de forma arbitraria una o más celdas de la red, y el número de canales disponible para la asignación.

5.1.1. Número de canales y celdas con asignación fija

En varios de los *benchmarks* propuestos como pruebas para los algoritmos es necesario asignar al menos una de las celdas de mayor demanda para lograr que la frecuencia convergencia sea diferente de 0% contando con el mínimo de canales para los que es posible solucionar el CAP.

Se encontró que es posible disminuir la cantidad de celdas con asignación fija o no asignar ninguna celda arbitrariamente aumentando el número de canales disponibles para la solución; de esta manera la optimización es de menor calidad considerando el problema de mínimo orden, sin embargo el algoritmo encuentra soluciones válidas con mayor autonomía.

5.1.2. Coeficientes A,B,C de la ecuación de movimiento

Como se trató en la sección 3.1 los términos ponderados por A y B en la ecuación de movimiento (19) representan las restricciones de demanda y compatibilidad electromagnética impuestas a la red respectivamente, mientras que el término C permite que se asignen canales de forma más agresiva en celdas cuya demanda no ha sido satisfecha. En este orden de ideas es claro como los coeficientes que ponderan estos términos inciden de forma directa en la velocidad o agresividad de la asignación de canales, haciéndola más o menos restrictiva.

Implementando la red con los pesos propuestos por Funabiki [7] (A=1, B=1, C=3,4 o 5), el número de canales asignados por celda varía de unos pocos canales a algunos cientos en iteraciones consecutivas. Esta variación se debe a que una alta ponderación del término de demanda, además del carácter paralelo de la red neuronal pueden provocar grandes cambios a la entrada de las neuronas. Se encontró que al darle un peso mayor al término B, que impone mayores restricciones a la red en comparación con los pesos A y C, disminuye la variación de canales asignados por celda entre iteraciones, lo que podría ayudar a la convergencia de la misma.

Sabiendo que para el problema P13 de Philadelphia puede lograrse una frecuencia de convergencia de 100% según los reportes de Funabiki [7] se propuso un análisis de dicho indicador variando los pesos de los términos A,B,C con valores discretos entre 20% y 100% con un número de tres celdas iniciadas de forma fija. Siendo que el valor de C varía de forma aleatoria entre 3,4 y 5, se definió un nuevo peso para el término C llamado PesoC, la ecuación de movimiento queda entonces dada por (23)

para $mod(t,T) < w$

$$DeltaU(i,j) = -A * f\left(\sum_{q=1}^{m} V(i,q) - V_{dem}(i)\right) - B\left(\sum_{\substack{q=j-(Mcomp\ (i,i)-1) \\ q\neq j \\ 1\leq q\leq n}}^{j+(Mcomp\ (i,i)-1)} V(i,q) + \sum_{\substack{p=1 \\ p\neq i \\ Mcomp\ (i,p)>0}}^{n} \sum_{q=j-(Mcomp\ (i,p)-1)}^{j+(Mcomp\ (i,p)-1)} V(p,q)\right) * V(i,j)$$

$$+PesoC * C * h\left(\sum_{q=1}^{m} V(i,q) - V_{dem}(i)\right) * \left(1 - V(i,j)\right)$$

para $mod(t,T) < w$

$$DeltaU(i,j) = -A * f\left(\sum_{q=1}^{m} V(i,q) - V_{dem}(i)\right) - B\left(\sum_{\substack{q=j-(Mcomp\ (i,i)-1) \\ q\neq j \\ 1\leq q\leq n}}^{j+(Mcomp\ (i,i)-1)} V(i,q) + \sum_{\substack{p=1 \\ p\neq i \\ Mcomp\ (i,p)>0}}^{n} \sum_{q=j-(Mcomp\ (i,p)-1)}^{j+(Mcomp\ (i,p)-1)} V(p,q)\right)$$

$$+PesoC * C * h\left(\sum_{q=1}^{m} V(i,q) - V_{dem}(i)\right) * \left(1 - V(i,j)\right)$$

(23)

Se presentan a continuación los resultados de frecuencia de convergencia para el problema P13, con tres celdas con asignación fija y para distintos valores de A, B y PesoC variando entre 0.2 y 1.

Tabla 3. Frecuencia de convergencia para P13 variando A, B, PesoC

Simulación	A	B	PesoC	Frecuencia de Convergencia (%)	Simulación	A	B	PesoC	Frecuencia de Convergencia (%)
1	1	1	1	0	33	0,5	1	1	0
2	1	1	0,8	0	34	0,5	1	0,8	0
3	1	1	0,5	0	35	0,5	1	0,5	6
4	1	1	0,2	8	36	0,5	1	0,2	84
5	1	0,8	1	0	37	0,5	0,8	1	0
6	1	0,8	0,8	0	38	0,5	0,8	0,8	0
7	1	0,8	0,5	0	39	0,5	0,8	0,5	1
8	1	0,8	0,2	4	40	0,5	0,8	0,2	71
9	1	0,5	1	0	41	0,5	0,5	1	0
10	1	0,5	0,8	0	42	0,5	0,5	0,8	0
11	1	0,5	0,5	0	43	0,5	0,5	0,5	0
12	1	0,5	0,2	2	44	0,5	0,5	0,2	22
13	1	0,2	1	0	45	0,5	0,2	1	0
14	1	0,2	0,8	0	46	0,5	0,2	0,8	0
15	1	0,2	0,5	0	47	0,5	0,2	0,5	0

16	1	0,2	0,2	0	48	0,5	0,2	0,2	0
17	0,8	1	1	0	49	0,2	1	1	2
18	0,8	1	0,8	0	50	0,2	1	0,8	10
19	0,8	1	0,5	13	51	0,2	1	0,5	85
20	0,8	1	0,2	9	52	0,2	1	0,2	96
21	0,8	0,8	1	0	53	0,2	0,8	1	0
22	0,8	0,8	0,8	0	54	0,2	0,8	0,8	2
23	0,8	0,8	0,5	0	55	0,2	0,8	0,5	74
24	0,8	0,8	0,2	2	56	0,2	0,8	0,2	94
25	0,8	0,5	1	0	57	0,2	0,5	1	0
26	0,8	0,5	0,8	0	58	0,2	0,5	0,8	0
27	0,8	0,5	0,5	0	59	0,2	0,5	0,5	19
28	0,8	0,5	0,2	1	60	0,2	0,5	0,2	98
29	0,8	0,2	1	0	61	0,2	0,2	1	0
30	0,8	0,2	0,8	0	62	0,2	0,2	0,8	0
31	0,8	0,2	0,5	0	63	0,2	0,2	0,5	0
32	0,8	0,2	0,2	0	64	0,2	0,2	0,2	75

Fuente: Presentación propia de los autores.

Se concluye que los valores que deben usarse en la simulación son A=0.2, B=0.5 y PesoC=0.2, para los cuales se encontró la mayor frecuencia de convergencia.

5.1.3. Resultados publicados por Funabiki

En su artículo, Funabiki [7] publicó resultados para ocho instancias. Estos escenarios comprenden seis *benchmarks* de Philadelphia, el escenario de Finlandia y un ejemplo sencillo de una red de cuatro celdas (F1). Se presenta una comparación de los resultados obtenidos por Funabiki y los obtenidos en el presente trabajo. Además de estos resultados, se presenta en la sección siguiente la frecuencia de convergencia obtenida para los trece *benchmarks* de Philadelphia restantes.

Tabla 4. Comparación de resultados obtenidos y publicados por Funabiki.

Escenario	Frecuencia de convergencia		Iteraciones promedio		Número de canales		Límite teórico de canales
	Publicada	Obtenida	Publicadas	Obtenidas	Publicado	Obtenido	
F1	100%	97%	21.2	29.65	11	11	11
P13	100%	100%	117.5	155.61	309	309	309
P14	100%	98%	117.5	103,29	533	533	533
P15	24%	92%	305.6	190.09	309	353	309
P16	100%	85%	100.3	225.83	533	533	533
P17	79%	77%	234.8	203,15	221	221	221
P18	93%	99%	147.8	106.6	381	381	381
P20	24%	31%	305.6	285.09	73	75	73

Fuente: Presentación propia de los autores.

5.1.4. Resultados Obtenidos

El algoritmo de Hopfield-Funabiki fue puesto a prueba para los 19 casos de Philadelphia(P1 a P19), el caso de Finlandia (P20) y el escenario sencillo de cuatro celdas planteado en [7]. Para las simulaciones de algunos escenarios se iniciaron hasta tres celdas.

Tabla 5. Resultados obtenidos para todas las instancias.

Escenario	ID Celda Iniciada	N. Canales	Frecuencia de convergencia	Iteraciones promedio	Mínimo teórico de canales
F1	-	11	27%	109,7	11
	4	11	97%	29,65	
P1	-	280	59%	292,27	229
P2	12	309	0%	-	257
	-	309	9%	402,55	
P3	-	533	21%	329,9	426
P4	-	1072	12%	349,66	855
P6	-	229	2%	385,5	179
	-	235	17%	306,23	
	-	240	40%	313,32	
P7	-	309	6%	338,5	252
P8	9	533	65%	292,95	426
P9	9,16	610	12%	400,26	524
P10	-	500	55%	412,03	426
P11	-	510	87%	291,98	426
	-	533	94%	254,19	
P12	8,9,16	533	92%	182,31	532
	9,16	533	2%	276	
P13	10,11,12	309	100%	155,61	308
	11,12	309	1%	160	
P14	8,9,16	533	98%	103,29	532
	9,16	533	98%	144,45	
	9	533	82%	218,48	
	-	533	0%	-	
P15	10,11,12	353	92%	190,09	308
	11,12	353	22%	235,9	
P16	9,16	533	85%	225,83	532
	9	533	21%	320,23	
	-	533	0%	-	
P17	10,11,12	221	77%	203,15	220
	11,12	221	0%	-	
	8,9,16	381	100%	72,44	
P18	9,16	381	99%	106,6	380

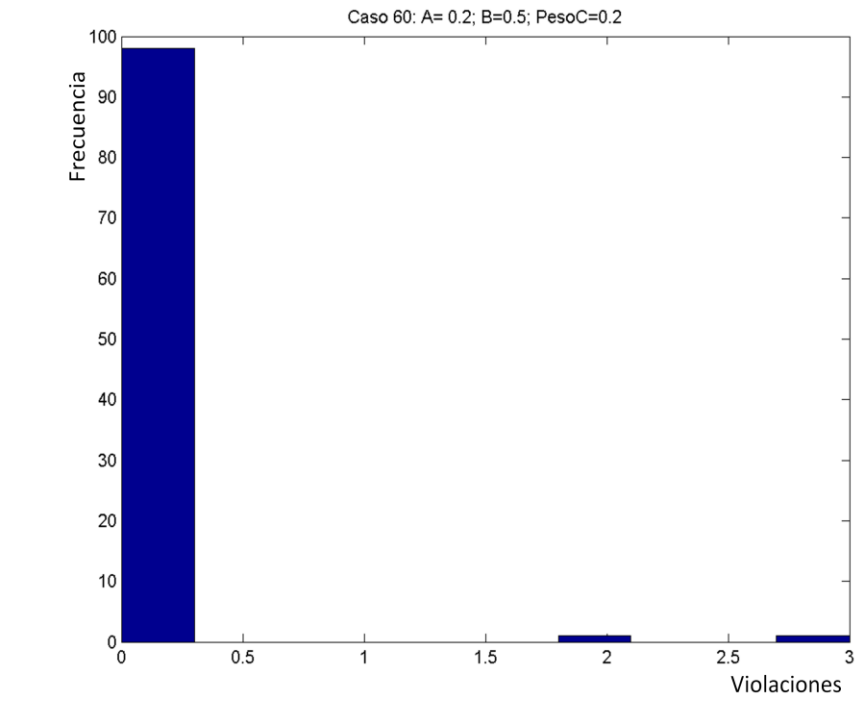
	9	381	66%	180,07	
	-	381		-	
P19	10,11,12	533	99%	187,02	528
	11,12	533	0%	-	
P20	-	75	31%	285,09	72
	-	80	99%	170,97	

Fuente: Presentación propia de los autores.

5.1.5. Pesos utilizados para la ecuación de movimiento

De los resultados obtenidos para algunas instancias se intuye que los pesos óptimos de los términos de la ecuación movimiento para una instancia dependen de la matriz de compatibilidad y son independientes del vector de demanda. Ver anexos donde se presentan los histogramas de la distribución del número de violaciones de la asignación resuelta con el algoritmo de Hopfield-Funabiki como histogramas para las instancias P13 y P18. El mejor resultado para la instancia P13 con asignación fija en tres celdas se presenta en la Figura 16.

Figura 16. Histograma caso 60



Fuente: Presentación propia de los autores.

5.2. Simulated Annealing

Tabla 6. Resultados algoritmo SA.

Escenario	ID Celda Iniciada	**Numero de Canales Simulación**	Minimo Teorico	**Frecuencia Convergencia SA**	Temperatura de Convergencia
F1	-	**11**	11	**100%**	5,33
P1	-	**260**	229	**90%**	2,29
P2	-	**300**	257	**100%**	3,61
P3	-	**533**	426	**100%**	4,7
P4	-	**1072**	855	**100%**	2,66
P5	-	**1765**	1719		
P6	-	**229**	179	**100%**	4,89
P7	-	**300**	252	**100%**	2,6
P8	9	**533**	426	**100%**	4,33
P9	-	**610**	524	**100%**	3,23
P10	-	**500**	426	**100%**	4
	-	**463**		**40%**	3,5
P11	-	**510**	426	**100%**	3,23
P12	9	**533**	532	**100%**	5,33
P13 (F7)	10,11,12	**309**	308	**100%**	6,17
P14	9	**533**	532	**100%**	5,17
P15	11	**353**	308	**100%**	2,95
P16	9	**533**	532	**100%**	3,83
P17	11	**221**	220	**100%**	1,69
P18	9	**381**	380	**100%**	5,67
P19	10,11,12	**533**	528	**100%**	6,5
P20	-	**73**	72	**100%**	3,68

Fuente: Presentación propia de los autores.

Se simularon las instancias que tuvieron una frecuencia de convergencia para el algoritmo de Hopfield-Funabiki en las mismas condiciones en que se hizo para este último, es decir el mismo número de canales y las mismas celdas con asignación fija.

Se presentan también resultados de simulaciones más agresivas donde el número de celdas con asignación fija así como el número de canales disponibles para la asignación es menor.

Los resultados obtenidos son de una frecuencia de convergencia de 100% para todas las instancias, esto muestra que la propuesta realizada es una mejora sustancial al algoritmo de Hopfield-Funabiki, al precio de un aumento sustancial del tiempo de cómputo.

La temperatura de Convergencia es una medida promedio de la temperatura con que finalizó el proceso de annealing para el algoritmo de SA, empezando en todas las simulaciones con un valor inicial de 10, y un valor de épsilon de 50.

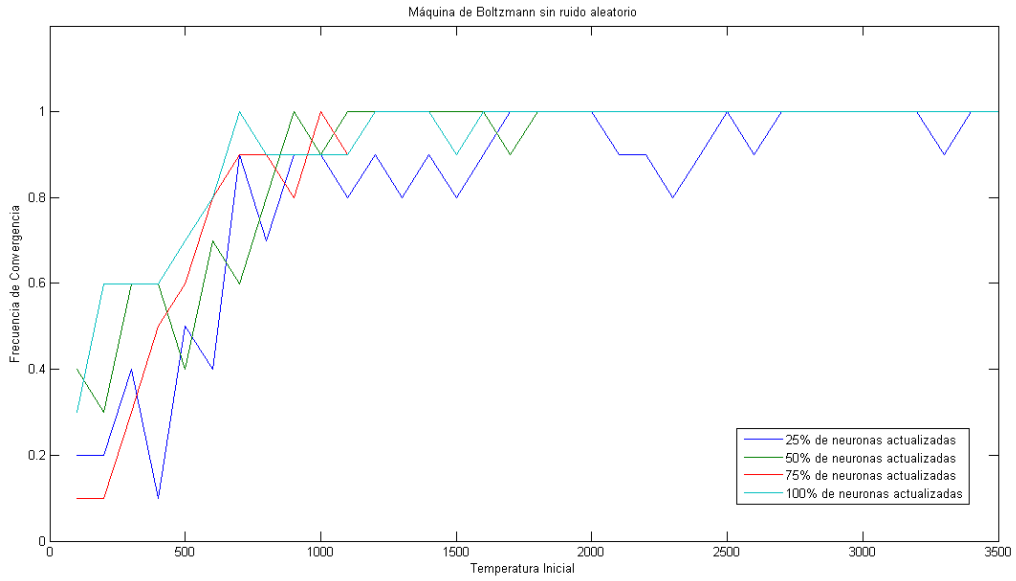
5.3. Máquina de Boltzmann (MB)

Se halló la frecuencia de convergencia de MB para el CAP en el escenario F1 para valores de temperatura inicial entre 100 y 10000 con un cambio de temperaturas de 100. El número de simulaciones de MB para calcular la frecuencia de convergencia para cada valor de temperatura inicial fue de 25.

Cambio en la frecuencia de convergencia al aumentar la temperatura inicial

De [4] sabemos que para un tiempo lo suficientemente grande de cómputo la frecuencia de convergencia de la máquina de Boltzmann debe llegar a ser de 100%. La cantidad de iteraciones para MB es directamente proporcional al valor de temperatura inicial del proceso de annealing. En la Figura 17 se muestra como la frecuencia de convergencia llega a su valor óptimo para temperaturas iniciales lo suficientemente altas. El porcentaje de neuronas actualizadas fue de 25%, 50%, 75% y 100%, correspondiendo cada curva de la gráfica a estos porcentajes.

Figura 17. Maquina de Boltzmann sin ruido aleatorio.

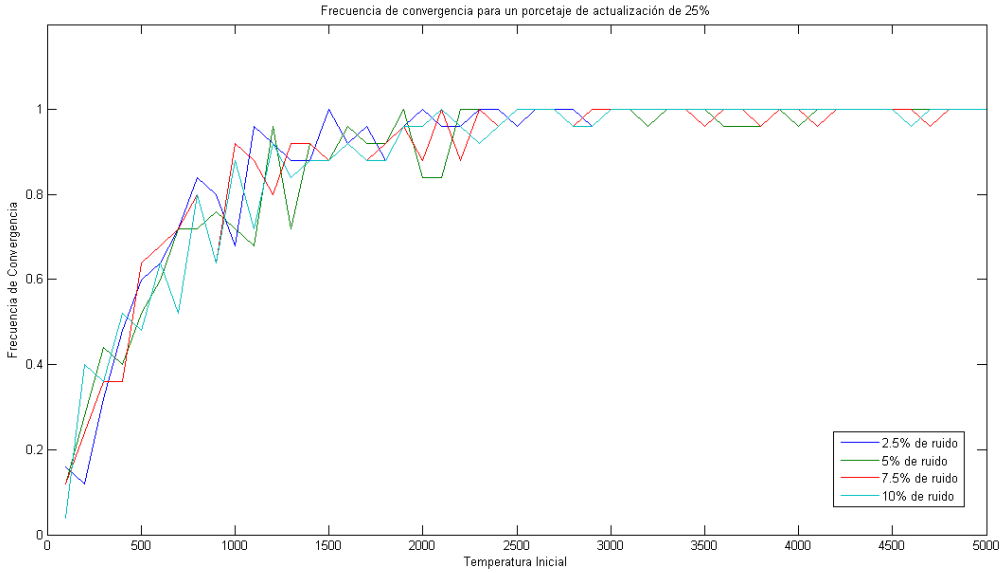


Fuente: Presentación propia de los autores.

Incidencia en la frecuencia de convergencia del porcentaje de neuronas actualizadas y de la cantidad de ruido aleatorio introducido en la red

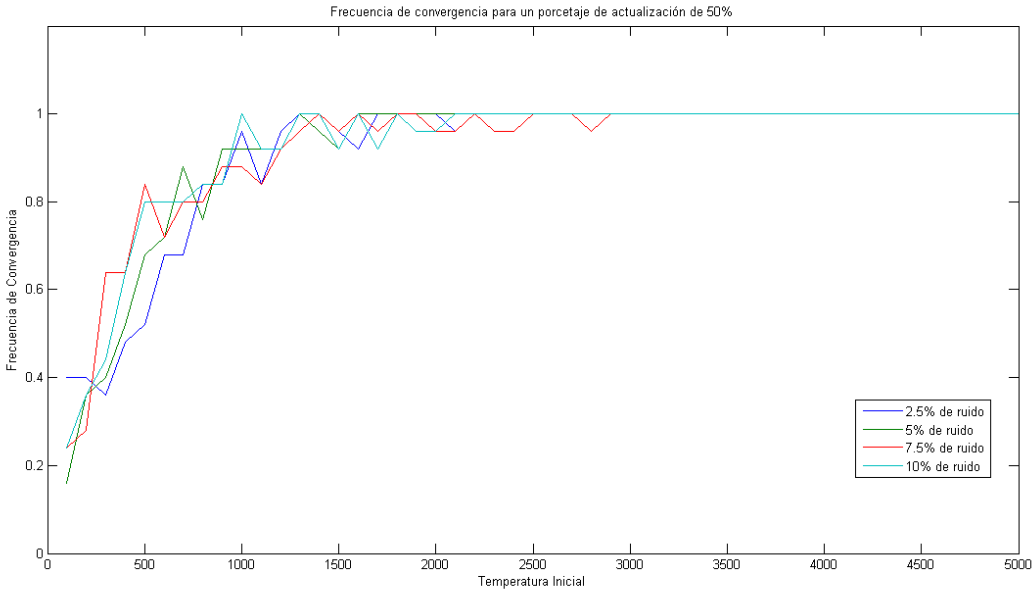
Se realizó la simulación de frecuencia de convergencia contra temperatura inicial introduciendo ahora porcentajes de ruido aleatorio en 2.5%, 5%, 7.5% y 10% de las neuronas de la MB. Las figuras Figura 18, Figura 19, Figura 20, y Figura 21 muestran el comportamiento de la frecuencia de convergencia para los porcentajes de ruido mencionados, fijando en cada gráfica la cantidad de neuronas actualizadas en 25%, 50%, 75% y 100%.

Figura 18. Frecuencia de convergencia para un porcentaje de actualización de 25%.



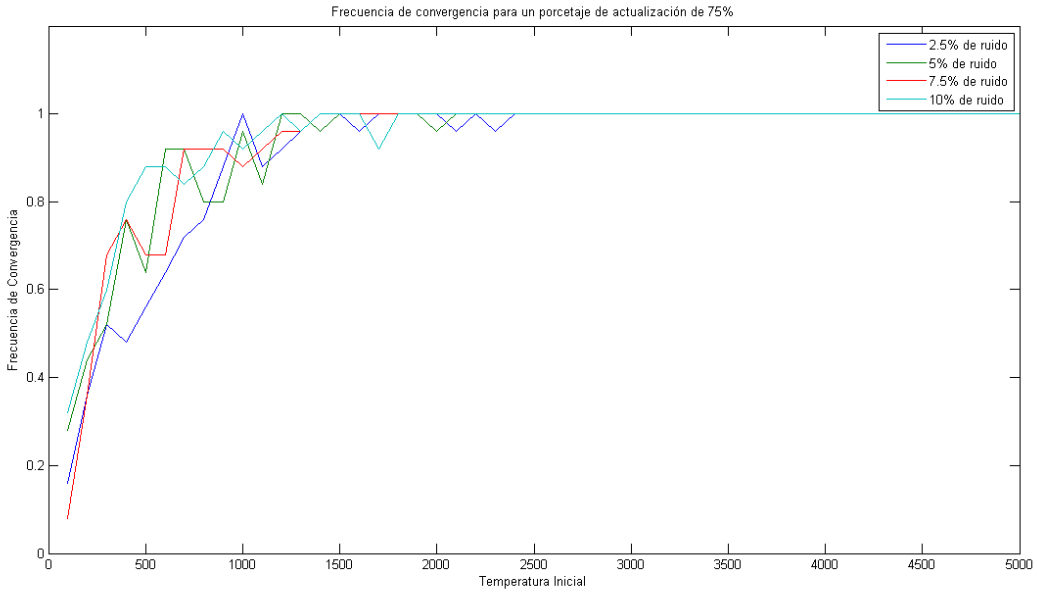
Fuente: Presentación propia de los autores.

Figura 19. Frecuencia de convergencia para un porcentaje de actualizacion de 50%.



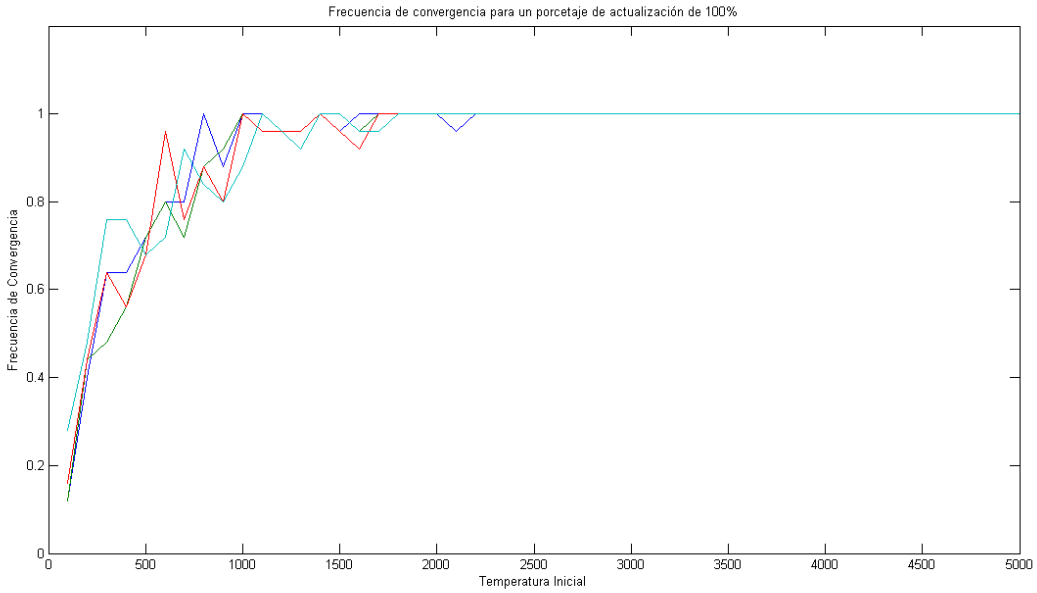
Fuente: Presentación propia de los autores.

Figura 20. Frecuencia de convergencia para un porcentaje de actualización de 75%.



Fuente: Presentación propia de los autores.

Figura 21. Frecuencia de convergencia para un porcentaje de actualización del 100%.



Fuente: Presentación propia de los autores.

Las gráficas para 50%, 75% y 100% de neuronas actualizadas en la red muestran claramente que para los valores más bajos de ruido introducido a la red se tienen frecuencias de convergencias menores a temperaturas iniciales bajas, es decir, que se logran mejores resultados con valores relativamente altos de ruido cuando el tiempo de cómputo es menor. En la gráfica para 25% de neuronas actualizadas el comportamiento no es claro.

Se observa progresivamente en las gráficas que a medida que el porcentaje de actualización aumenta, la frecuencia de convergencia crece con mayor rapidez, es decir, se llega a su valor óptimo a temperaturas iníciales más bajas.

5.3.1. Simulación de Instancias de Philadelphia y Finlandia

Tabla 7: Resultados Máquina de Boltzmann

Instancia	No. De canales	Violaciones				T_inicial
		Demanda	Cosite	Adj_Ch	Totales	
F1	11	0	0	0	0	15000
P1	280	10	0	262	272	15000
P2	309	10	12	316	338	15000
P3	533	5	28	260	293	15000
P4	1072	3	176	846	1025	15000
P6	240	15	0	222	237	15000
P7	209	8	8	212	228	15000
P8	533	4	40	228	272	15000
P9	610	5	22	272	299	15000
P10	500	4	32	340	376	15000
P11	533	0	74	268	342	15000
P12	533	9	20	246	275	15000
P13	309	11	0	234	245	15000
P14	533	9	14	182	205	15000
P15	353	15	0	200	215	15000
P16	533	12	8	206	226	15000
P17	221	11	0	286	297	15000
P18	381	12	10	296	318	15000
P19	533	15	0	182	197	15000
P20	80	14	0	26	40	15000

Fuente: Presentación propia de los autores

6. CONCLUSIONES

El algoritmo de Hopfield-Funabiki logró encontrar soluciones óptimas al CAP en varios de los casos de prueba. Sin embargo, para asegurar su convergencia fue necesaria en varias instancias la asignación fija de una o más celdas. Se descubrió que los coeficientes que ponderan los términos de la ecuación de movimiento están estrechamente relacionados con la frecuencia de convergencia, y constituyen por tanto una importante oportunidad de optimización para este algoritmo.

Se obtuvo una mejora significativa al combinar una red neuronal tipo Hopfield con el método de Simulated Annealing, propuesta con la cual se alcanzó un 100% de convergencia para la mayoría de instancias de prueba con una cantidad igual o menor de canales disponibles para el CAP. Esta propuesta es novedosa y puede ser extendida a diferentes criterios de optimización, al costo de un mayor esfuerzo computacional.

La máquina de Boltzmann es conocida por su baja velocidad de convergencia. Sin embargo, a diferencia de otros tiende a mejores resultados a medida que aumenta la temperatura inicial, o lo que es equivalente el tiempo de simulación. Se concluyó que la cantidad de neuronas actualizadas en una versión secuencial de la misma así como el porcentaje de ruido introducido a la red inciden en la frecuencia de convergencia. Para instancias pequeñas se obtuvieron resultados mejores que en métodos con mayor velocidad de convergencia, lo que alienta a futuros trabajos que extiendan los alcances de este algoritmo.

En el presente trabajo de grado se logró exitosamente unificar una serie de benchmarks internacionales en un formato matricial uniforme, facilitando así la simulación de dichos escenarios con los algoritmos planteados. Aunque podría parecer simple, este avance es muy significativo ya que en muchas ocasiones estos benchmarks se encuentran en formatos confusos que dificultan su implementación.

7. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN

[1] Hertz, J.; Krogh, A.; Palmer, R.G.; "Introduction to the theory of neural computation" *Addison-Wesley Publishing Company.* Capítulo 1, Apr 1995.

[2] Hilera, G.; Jose, R.; "Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones", *Addison-Wesley Iberoamericana*. 1995.

[3] Freeman, J.; Skapura, D.; "Redes neuronales: algoritmos, aplicaciones y técnicas de programación" *Addison-Wesley/ Diaz de Santos.* Capitulo 4, 1993.

[4] Thuijsman, F.; Weijters, A.; "Artificial neural networks: an introduction to ANN theory and practice" *Springer*. Pág. 119 a 128, 1995.

[5] Flores, R.; Fernández, J.; "Las Redes Neuronales Artificiales: Fundamentos teóricos y aplicaciones prácticas" *Netbiblo.* Sección 3.6, 2008.

[6] Catharinus, A.K.; "Frequency Assigment: Models and Algorithms" *Arie M.C.A. Koster. Proefschrift Universitet Maastricht,* Nov. 1999.

[7] Funabiki, N.; Takefuji, Y., "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Transactions on Vehicular Technology*, Vol.41, No.4, Nov. 1992.

[8] Kunz, D.; "Channel Assignment for Cellular Radio Using Neural Networks", *IEEE Transactions on Vehicular Technology*, Vol.40, No.1, Feb. 1991.

[9] Santiago, R.C.; Gigi, E.; Lyandres, V., "An improved heuristic algorithm for frequency assignment in nonhomogeneous cellular mobile networks," *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, Vol.7, Sept. 2004.

[10] Aardal, K.; Van Hoesel, S.; Koster, A.; Mannino, C.; Sassano, A.; "Models and Solutions Techniques for Frequency Assignment Problems" *Konrad-Zuse-Zentrum fur Informationstechnik*, 1999.

[11] Smith, K.; Palaniswami, M.; "Static and dynamic channel assignment using neural networks," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 2, Feb. 1997.

[12] Salcedo-Sanz, S.; Santiago-Mozos, R.; Bousono-Calzon, C., "A hybrid Hopfield network-simulated annealing approach for frequency assignment in satellite communications systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 34, No. 2, Apr. 2004.

[13] VanLaarhoven, P.; Aarts, E.; "Simulated Annealing: Theory and applications" *Kluwer Academic Publishers,* Capitulo 2, 1987.

[14] (2010) Konrad-Zuse-Zentrum für Informationstechnik Berlin website [en línea]. Disponible: http://fap.zib.de

[15] Chandy, J.; Sungho K.; Ramkumar, B.; Parkes, S.; Banerjee, P.; "An evaluation of parallel simulated annealing strategies with application to standard cell placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.16, No.4, Apr. 1997.

[16] Aarts, E.; Korst, J.; "Boltzmann Machine as a Model for parallel Annealing" *Algorithmica*, 1991.

[17] Tesar, B.; Kapenga, J.; Trenary, R.; "A Boltzamnn machine solution of the traveling salesperson problem: A study for parallel implementation" *Computer Science Department, Western Michigan University, USA*, 1989.