

# Optimización dinámica posibilística de un sistema de distribución mediante una hiperheurística basada en razonamiento con dominio en metaheurísticas x-pso multi-objetivo

Dynamic Optimization of a Distribution System by a Case Based Reasoning Hyperheuristic With X-Pso Multi-Objective Domain

Gustavo Schweickardt<sup>a\*</sup>, Carlos Casanova<sup>a</sup>.

Recibido: septiembre 3 de 2014  
Recibido con revisión: mayo 19 de 2015  
Aceptado: mayo 28 de 2015

<sup>a\*</sup>Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay, Ing. Pereira 676 - 3260, Concepción del Uruguay, Argentina  
Tel.: +(54) 3442 423898  
gustavoschweickardt@conicet.gov.ar  
casanovac@frcu.utn.edu.ar

Energética 45, junio (2015), pp. 83-93

ISSN 0120-9833 (impreso)  
ISSN 2357 - 612X (en línea)  
www.revistas.unal.edu.co/energetica  
© Derechos Patrimoniales  
Universidad Nacional de Colombia



## RESUMEN

En el presente trabajo se aplica una Hiperheurística que los autores han denominado HY X-FPSO CBR para la Optimización Dinámica Posibilística que se corresponde con la Planificación de Mediano/Corto Plazo de un Sistema de Distribución de Energía Eléctrica. El problema a resolver, específicamente, es la definición/identificación del Espacio de Estados por los que el Sistema de Distribución de Energía Eléctrica debe evolucionar, conociendo el número de etapas. Las mismas se corresponden con cada año del Período de Control Regulatorio. Se procede al Diseño y Entrenamiento de la Red Neuronal Artificial de Retropropagación, en la que se basa el mecanismo de Aprendizaje/Selección mediante el cual son aplicadas las formas X-FPSO, para cierta instancia de decisión, identificadas como dominio de la Hiperheurística propuesta. Se integran los resultados al Modelo de Optimización Posibilística y se procede a la simulación completa en un Estudio de Caso sobre un Sistema de Distribución de Energía Eléctrica real. Se comparan los resultados que arroja la Hiperheurística propuesta, con el Espacio de Estados formado por inspección, en otras optimizaciones realizadas sobre el mismo Sistema de Distribución de Energía Eléctrica.

## PALABRAS CLAVE

Optimización; Enjambre de Partículas; Hiperheurísticas; Redes Neuronales Artificiales; Sistema de Distribución de Energía Eléctrica

## ABSTRACT

In this work a Hyperheuristic HY X-PSO CBR, supporting a Possibilistic Dynamic Optimization corresponding to the Electric Distribution System Planning in the Mid/Short Term, is applied. The problem to solve is the definition/identification of States Space for the evolution of Electric Distribution System, knowing the number of stages, corresponding with each of years of Regulatory Control Period. The Artificial Backpropagation Neural Network, that support the Selection/Learning Method to choice, in certain decision instance, the X-FPSO form in Metaheuristics domain of proposed Hyperheuristic, is Designed and Trained. The results are used in a Dynamic Possibilistic Model on a real Electric Distribution System, as Study Case. The Hyperheuristic results and States Space defined by mean inspection for optimizations performed on the same real Electric Distribution System, are compared.

## KEYWORDS

Optimization; Particles Swarm; Hyperheuristics; Artificial Neural Networks; Electric Distribution System.

## 1. INTRODUCCIÓN

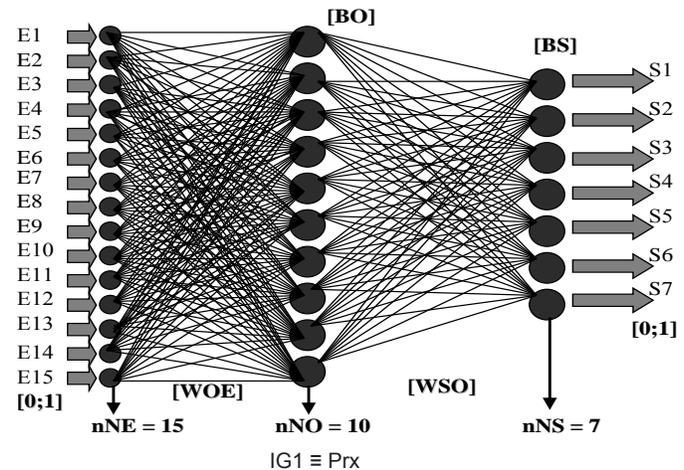
Uno de los problemas más complejos en la Optimización Dinámica en general, y que en este trabajo se corresponde, particularmente con la Planificación de Mediano/Corto Plazo requerida a los fines de Control Regulatorio, sobre un Sistema de Distribución de Energía Eléctrica (SDEE), es la identificación/definición del Espacio de Búsqueda, asociado a su trayectoria óptima de evolución. Como las etapas están definidas, ya que cada una se corresponde con un año corte del período de control, el problema se circunscribe a la identificación/definición del Espacio de Estados por los que el sistema puede evolucionar, con mayor o menor aptitud. Si a esta descripción, se le incorporan aspectos realistas, tales como la Dinámica Posibilística requerida y la Optimización Multi-Objetivo, cuestiones tratadas por los autores en otros trabajos [Schweickardt & Miranda, 2009], la complejidad del problema aumenta notoriamente. Hasta el momento, en los trabajos referidos, el Espacio de Estados ha quedado definido por inspección, tarea, de hecho, laboriosa y aproximativa. Habida cuenta del esfuerzo requerido y la ausencia metodológica para resolver tal problema, se abordó la investigación de un método para su solución, arribando al desarrollo de la Hiperheurística HY X-FPSO CBR. Cabe la aclaración, redundante tal vez: la extensión de la Metaheurística PSO canónica Mono-Objetivo al dominio Multi-Objetivo, es sustentada en el empleo de Conjuntos Difusos (*Fuzzy Sets*) solidarios a cada objetivo/restricción del problema abordado. Por ello en vez de referir el concepto: forma X-PSO Multi-Objetivo, se refiere: forma X-FPSO. En el presente trabajo se sigue la secuencia de pasos, brindando los desarrollos específicos, para la aplicación de la Hiperheurística propuesta en la definición del Espacio de Estados, por los que evolucionará el SDEE según la trayectoria más satisfactoria resultante de una Optimización Dinámica Posibilística. Se presentan, finalmente, los resultados más relevantes obtenidos, sin proporcionar desarrollos específicos sobre tal optimización, ya que coincide en un todo con la presentada por los autores en [Schweickardt & Miranda, 2009].

## 2. LA RED NEURONAL DE RETROPROPAGACIÓN PARA APROXIMAR LA FUNCIÓN DE APRENDIZAJE/SELECCIÓN DE LA HIPERHEURÍSTICA EN EL DOMINIO DE LAS METAHEURÍSTICAS X-FPSO

### 2.1. Redes Neuronales de Retropropagación (RNR)

El tipo de Red Neuronal (se omite la palabra *artificial*) de Retropropagación o *Backpropagation*, RNR, puede

clasificarse como una Red Neuronal de Entrada Continua y de Aprendizaje Supervisado. Un esquema de la misma, en su Modelado de Capas-Neuronas, se puede apreciar en la Figura 1, que representa el diseño final de la RNR solidaria a la Hiperheurística propuesta, y cuyos elementos serán explicados cuantitativamente y con mayor detalle en el epígrafe que sigue.



**Figura 1:** Esquema de la Red Neuronal de Retropropagación (RNR) aplicada como Método de Aprendizaje/Selección de la HY X-PSO CBR

**Fuente.** Elaboración Propia

Respecto del concepto de *retropropagación*, no se proporcionarán detalles aquí, sino descriptivos, por existir abundante bibliografía para consultar en términos de diseño y algoritmos de aprendizaje o entrenamiento de la red, por caso: [Widrow & Lehr, 1990; Simpson, 1992; Funahashi & Nakamura, 1993; Hilera & Martínez, 1995; Hagan et. al., 1996].

Para el esquema de la Figura 1, se tiene una *capa de entrada*, una *capa oculta* y una *capa de salida*, con sus *neuronas* correspondientes, representadas por círculos. Pueden existir *diseños multicapa* que empleen *mayor número de capas ocultas*. Su número es *indeterminado* y depende: del problema a resolver, de la precisión en el algoritmo de entrenamiento empleado, del número de neuronas ocultas en cada una de estas capas (también indeterminado), y de la característica de procesamiento o función de transferencia asociada a cada neurona, entre otros aspectos. El *diseño de la mejor RNR*, si bien existen recomendaciones *acopiadas desde la experiencia*, tiene una *importante componente de prueba y error*.

En la capa de entrada se aplica el vector de entrada  $[E]$  (vector columna cuyo número de filas es el número de entradas) el cual, para cierta instancia de sus componentes, constituye un patrón de entrada. Entre la capa de entrada y la capa oculta (siempre refiriéndose a la Figura 1, existen conexiones o sinapsis, que van desde cada neurona de la capa de entrada, a cada neurona de la capa de oculta, tal y como se representa mediante un trazado de líneas. Lo mismo ocurre entre la capa oculta y la capa de salida, en donde, por propagación del patrón de entrada, es obtenido el

vector de salida  $[S]$  (vector columna cuyo número de filas es el número de salidas). El mismo, para instancia de sus componentes, correspondiente con la entrada, se constituye en el patrón de salida resultante. Considerando las conexiones o sinapsis entre las neuronas de la capa oculta y de la capa de entrada, son caracterizadas por un valor llamado peso sináptico, que será indicado por la variable real genérica  $w_{oeij}$  ( $w_{oeij} \equiv$  peso), aplicada entre la neurona oculta  $i$ -ésima y la neurona de entrada  $j$ -ésima. Por ello, aparece una Matriz de Pesos Sinápticos entre dichas capas, que se indica como  $[W_{OE}]$ . Su orden corresponde a  $(nNO \times nNE)$ , siendo  $nNO$  el número de neuronas de las capas de entrada y oculta, respectivamente. De forma idéntica ocurre entre las sinapsis de las neuronas de la capa oculta y las de la capa de salida, teniéndose la variable real genérica que indica el peso sináptico  $w_{soki}$ , entre la neurona de la capa de salida  $k$ -ésima y la neurona de la capa oculta  $i$ -ésima, y una Matriz de Pesos Sinápticos indicada como  $[W_{SO}]$ . Su orden corresponde a  $(nNS \times nNO)$ , siendo  $nNS$  el número de neuronas de la capa de salida. Adicionalmente, puede incluirse una entrada independiente en cada neurona, la cual es referida como ganancia o bias. Dependiendo del diseño, puede o no intervenir. En la Figura 1, se tienen ganancias o biases para las neuronas de la capa oculta y las de la capa de salida. Son indicadas mediante las variables reales  $b_j$  y  $b_k$ , y sus vectores columna como  $[BO]$  y  $[BS]$ , respectivamente.

La RNR emplea un ciclo *propagación y adaptación* (de dos fases):

**1ra)** Se aplica un patrón a la entrada de la red como estímulo; éste se propaga desde la primera capa a través de las capas internas de la red, hasta generar un patrón de salida en la capa homónima. Luego, el patrón de salida obtenido es comparado con el patrón de salida establecido o esperado, y se calcula un error para cada una de las salidas obtenidas individualmente;

**2da)** Los errores en dichas salidas se propagan hacia atrás (desde aquí que el Algoritmo de Entrenamiento de la RN reciba el nombre de Retropropagación) partiendo de la capa de salida, hacia todas las neuronas de la capa oculta precedente, que contribuyen en forma directa a la salida resultante. Sin embargo, las neuronas de dicha capa oculta sólo reciben una fracción del error total, posible de cuantificar basándose, aproximadamente, en la contribución relativa que haya aportado cada una de estas neuronas a la salida obtenida. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una componente del error que describa su contribución relativa al error total. Conforme resulte la componente de error percibida, se actualizan los pesos sinápticos de conexión y ganancias asociados a cada neurona, entre capas contiguas, a los efectos de lograr, iterativamente, que la red converja hacia un estado que permita clasificar, correctamente, todos los patrones de entrenamiento, generando la salida esperada. La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias u ocultas, se auto-organizan de modo tal que “aprenden” a reconocer distintas características inherentes al espacio de patrones de entrada, no “observadas” con anterioridad. El proceso de adaptación de los pesos sinápticos (y ganancias o biases, si procede), se completa, iterativamente, cuando al presentarse el conjunto previsto de patrones de entrada, la RNR responde a cada uno con patrones

de salida, conteniendo errores individuales cuya suma (error global), se mantiene por debajo de cierto error pre-establecido.

Una vez completo el aprendizaje o entrenamiento de la RNR, toda vez que se le presente un patrón arbitrario de entrada, el cual que contenga “ruido” o esté incompleto, las neuronas de la(s) capa(s) oculta(s) de la red, responderán con una salida activa, si tal patrón se asemeja a las características que, individualmente, las neuronas hayan aprendido a reconocer durante su entrenamiento. Y a la inversa: las neuronas de la(s) capa(s) oculta(s) tendrán la tendencia a inhibir su salida, si el patrón de entrada no contiene característica alguna que puedan, individualmente, reconocer.

Varias investigaciones han demostrado que, durante el proceso de entrenamiento, la RNR tiende a desarrollar relaciones internas entre neuronas con el fin de organizar los datos de entrenamiento, según clases. Una red “entrenada” ha encontrado una representación interna tal, que le permite generar las salidas deseadas, cuando se les presentan las entradas previstas en los patrones durante el proceso de entrenamiento. Esta misma representación interna, se puede aplicar a entradas que la red no reconozca, clasificándolas conforme las características que compartan con los patrones de entrada utilizados en el entrenamiento. Desde allí que la RNR, pueda comportarse como un aproximador universal de funciones, que puede interpolar resultados ante estímulos no observados en su entrenamiento, por las asociaciones desarrolladas entre las sinapsis de sus capas ocultas, en general.

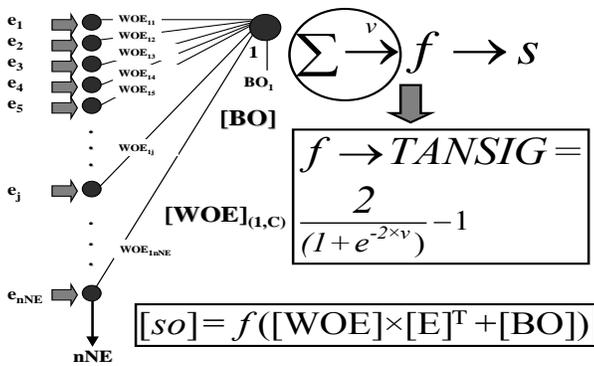
De modo que, por caso y volviendo sobre la Figura 1, *el resultado del entrenamiento de la RNR, serían las Matrices  $[W_{OE}]$ ,  $[BO]$ ,  $[W_{SO}]$  y  $[BE]$ .*

## 2.2. Diseño de la RNR para la HY X-FPSO CBR

Procede, previamente a presentar los detalles de diseño, explicar, de manera breve, el *funcionamiento de la neurona como un procesador elemental*, en el contexto de la RNR empleada por la Hiperheurística.

En la Figura 2, se aprecia la capa de entrada, en la que se presenta el patrón de entrada dado por  $[E]$ , cuyas componentes se indican mediante  $e_j$  con  $j$  en  $[1..nNE]$ .

Para la capa oculta, contigua, se indica sólo la neurona oculta 1, de modo que las conexiones sinápticas se presentan entre cada neurona de entrada y la neurona 1. En rigor, la neurona de entrada no procesa el estímulo recibido, sino que lo propaga, de manera que, en muchas referencias bibliográficas, son referidas como entradas.



**Figura 2:** Funcionamiento de la Neurona como Procesador Elemental: Función de Transferencia TANSIG empleada en el Diseño de la RNR

**Fuente.** Elaboración propia

Aquí se preserva la identificación de neurona, no obstante no constituya un procesador elemental, como se verá, cosa que sí es la neurona 1 de la capa oculta bajo análisis. Así queda conformado un vector de la 1er columna en la Matriz  $[WOE]$ , el cual se indica mediante  $[WOE]_{(1,C)}$ , puesto que todos los pesos sinápticos considerados, son de la forma  $woe_{1j}$ . Adicionalmente, se observa, desde el vector de ganancias o biases  $[BO]$ , el primer componente,  $bo_1$ , ganancia de la neurona oculta 1, como entrada adicional. Esto significa que existirá, sobre la neurona oculta 1, una entrada neta, proveniente de la propagación de los estímulos desde las neuronas de entrada, a través de las conexiones sinápticas y ponderadas por sus pesos sinápticos correspondientes, adicionando su ganancia. La composición de tal entrada neta, referida como  $v_1$ , tiene como expresión:

$$v_1 = \sum_{j=1}^{nNE} (woe_{1j} x e_j) + bo_1 \quad (1)$$

Por ello se indica en la Figura 2, entre un círculo, el símbolo sumatoria, que genera la entrada compuesta genérica  $v$ , conforme la regla establecida por la ecuación (1). Luego, la entrada,  $v$ , ingresa a lo que se refirió como Función de Transferencia en el epígrafe anterior, que se indica como  $f$ . Existen variadas Funciones de Transferencia que pueden ser empleadas, pero hay tres características funcionales predominantemente utilizadas en las RNR:

*PureLin:*  $f(v) = v$  (2)

*LogSig:*  $f(v) = 1/(1+e^{-v})$  (3)

*TanSig:*  $f(v) = (e^v - e^{-v})/(e^v + e^{-v}) = (2/(1 + e^{-2xv})) - 1$  (4)

Se observa que se ha empleado, para componer la salida  $so_1$  de la neurona oculta 1, la función *TanSig*, resultando:

$$so_1 = \text{TanSig}(v_1) = (2/(1 + e^{-2x(\sum_{nNE} (woe_{1j} x e_j) + bo_1)})) - 1 \quad (5)$$

Entonces, para el patrón de entrada  $[E]$  completo, y extendiendo (5) a todas las neuronas de la capa oculta, se tendrá el vector de salidas  $[so]$ , mediante la expresión:

$$[so] = \text{TanSig}([WOE] x [E]^T + [BO]) \quad (6)$$

Que se aprecia en la parte inferior derecha de la Figura 2, utilizando  $f$  en lugar de *TanSig*.

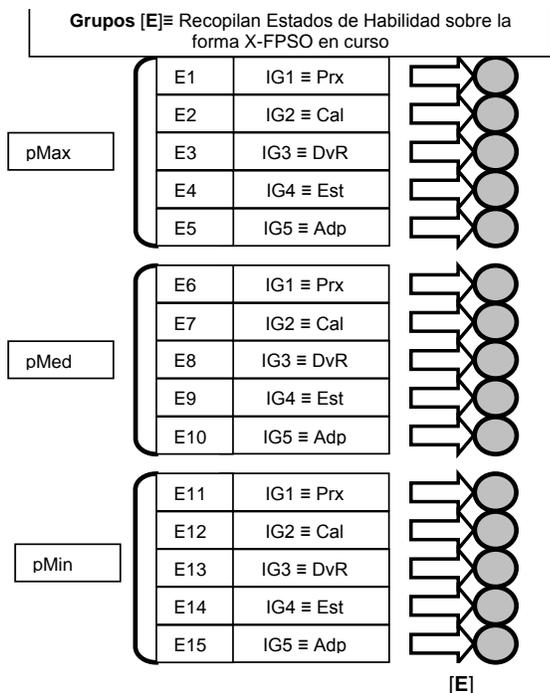
En general, se pueden combinar las Funciones de Transferencia. Para el diseño de la RNR soporte de la Hiperheurística propuesta, se ha empleado  $f = \text{TanSig}$  para formar el vector  $[so]$  y  $f = \text{PureLin}$  para formar el vector  $[S]$ , que proporciona cada patrón de respuesta o salida ante cada instancia de  $[E]$ . Este diseño de procesamiento, sumado a la estructura de una capa oculta, con  $nNO = 10$  neuronas ocultas, resultó ser el más adecuado luego de múltiples ensayos con otras estructuras. Se observa en la Figura 1, que las capas de entrada y de salida tienen 15 y 7 neuronas, respectivamente. Esta es la parte del diseño que se relaciona con los desarrollos teóricos de la HY X-FPSO CBR.

La propuesta, en tal sentido, es la siguiente:

**Diseño de la Capa de Entrada:** se tienen 15 entradas, cuyos valores son binarios (0 o 1). Cada grupo de 5, refiere a cierta partícula del enjambre, comandada en el espacio de soluciones por la forma X-FPSO que está siendo aplicada, en la instancia de decisión que se considera, desde la Hiperheurística. En las 5 entradas de cada grupo, se establece la pérdida (entrada 0) o no (entrada 1) de la habilidad que la partícula asociada tiene sobre la satisfacción de sendos Principios de la Inteligencia de Grupo (IG). Para ello, se consideran los elementos expuestos en los desarrollos teóricos de la HY X-FEPSO CBR. En tal contexto, son elegidas 3 partículas de la población o enjambre: la de mejor aptitud, la de aptitud promedio, y la de peor aptitud. Debido a las características que exhibe el Espacio de Búsqueda para la Clase de Problemas abordados, interpretado cada uno de ellos como una Optimización Mono-Objetivo (por ejemplo: minimización de pérdidas activas en alimentadores de Media Tensión del SDEE), cada partícula es suficientemente representativa. Posiblemente, si se eleva el número de partículas, seleccionadas de otro modo, podría obtenerse una mayor información. No obstante, desde las simulaciones efectuadas, esta configuración simple, resultó suficientemente representativa, en términos de la diversidad de localizaciones en el Espacio de Búsqueda. El orden de aptitud de las partículas, se corresponde con el índice decreciente de las entradas, al igual que, en cada grupo de 5 entradas, lo hacen los Principios de IG, conforme fueran presentados en los desarrollos teóricos. Se repiten a continuación:

**1) Proximidad (Prx); 2) Calidad (Cal); 3) Diversidad de Respuesta (DvR); 4) Estabilidad (Est) y 5) Adaptación (Adp).**

En la Figura 3, se representa la estructura del patrón de entrada así concebido, y su relación con la aptitud de la partícula y Principios de IG, según lo dicho.  $pMax$ ,  $pMed$  y  $pMin$ , son los acrónimos para las partículas de aptitud Máxima, Media y Mínima, respectivamente, del enjambre, en la instancia de decisión evaluada.  $IG_i$ , con  $i$  en  $[1..5]$ , son los Principios de IG y  $E_i$  las entradas binarias  $\{0, 1\}$ .



**Figura 3:** Diseño de la Capa de Entrada para la RNR de la HY X-FPSO CBR.

Fuente. Elaboración propia

*Diseño de la Capa de Salida:* se tienen 7 salidas, cuyos valores son también binarios (0 o 1). Estas 7 salidas, son divididas en dos grupos: el Grupo 1 se compone de las 5 primeras salidas, y el Grupo 2 de las 2 últimas. La representación puede apreciarse en la Figura 4.

En el Grupo 2, se tienen dos bits, de modo que existen 4 representaciones decimales posibles,  $Dec \equiv \{0, 1, 2, 3\} \rightarrow Dec+1 \equiv \{1, 2, 3, 4\} \equiv$  conjunto de números que constituyen el índice de cada una de las 4 X-FPSO formas, que componen el dominio de la Hiperheurística  $\equiv \{X1-FPSO, X2-FPSO, X3-FPSO, X4-FPSO\}$ .

Como se propuso en los desarrollos teóricos de la HY X-PSO CBR, las Metaheurísticas que componen tal dominio son:

**X1-FPSO:** PSO con Función de Decaimiento Inercial Multi-Objetivo

**X2-FPSO:** PSO con Factor de Constricción Multi-Objetivo

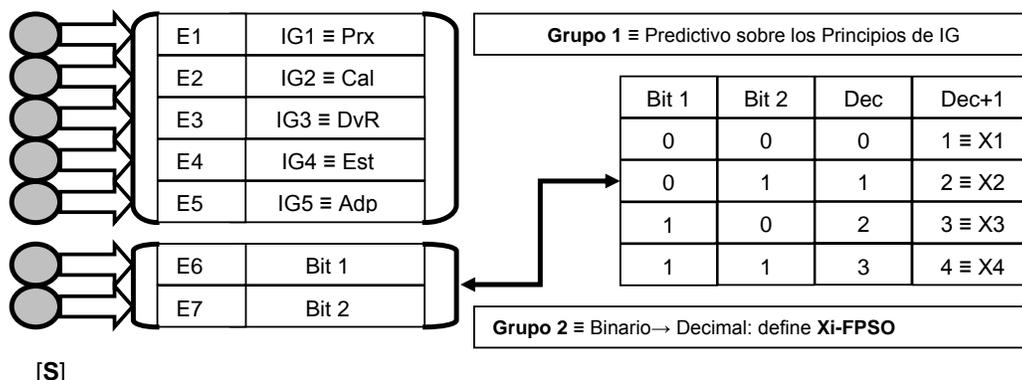
**X3-FPSO:** EPSO Multi-Objetivo

**X4-FPSO:** EPSO GIST Multi-Objetivo

En el Grupo 1, cada salida se corresponde con uno de los Principios de IG, idénticamente al diseño de la *capa de entrada* y en el mismo orden. Un valor 1 en cierta salida de este grupo, *supone la predicción* de que la aplicación de la forma X-PSO seleccionada e indicada por el Grupo 1, mejore el Principio de IG correspondiente; un valor 0 *supone la predicción* de que sostenga la habilidad sobre el Principio de IG correspondiente, o la mejore. La información del Grupo 1, es complementaria. Se capta a los efectos de reforzar la salida del Grupo 2, que selecciona la forma X-FPSO a emplear, con un componente heurístico, para el caso de que la RNR no pueda resolver, adecuadamente, un patrón de entrada no observado en el entrenamiento.

### 2.3. Entrenamiento de la RNR para la HY X-FPSO CBR. Aplicación de una MatHeurística Backpropagation con Gradiente Descendente y Momento, complementada con una Metaheurística PSO con Factor de Constricción.

El Algoritmo *Backpropagation* (de entrenamiento de la RNR) Clásico, se sustenta en una variante del denominado Método LMS (*Least Mean Square*) que *minimiza el error cuadrático medio global* de la RNR. Tal algoritmo fue introducido por Bernard Widrow, y su historia y desarrollo puede ser consultado en [Widrow & Lehr, 1990]. De las *variantes* más empleadas, disponibles en paquetes software para diseño, entrenamiento y simulación de RNA (por caso, MATLAB® y sus aplicaciones de tercera parte), sobre las que no se brindarán detalles matemáticos, se tienen:



**Figura 4:** Diseño de la Capa de Salida para la RNR de la HY X-FPSO CBR.

Fuente. Elaboración Propia

A) Backpropagation Base o de Gradiente Descendente:

Constituye la más simple implementación de este método de aprendizaje, y su forma consiste en la actualización iterativa de los Pesos Sinápticos y Ganancias o Biases, en la dirección en la cual la función error cuadrático medio global decrece más rápidamente.

B) Backpropagation según Gradiente Descendente con Momento:

Constituye una variante de A), en la cual se incorpora un parámetro denominado coeficiente de Momento (o Momentum). El mismo permite a la red ignorar puntos característicos de la superficie de la función error cuadrático medio global, en las que el entrenamiento podría quedar “atrapado”, arrojando un mínimo local.

C) Backpropagation con Razón de Aprendizaje Variable:

Constituye una variante del Método del Gradiente Descendente con o sin Momento, en la cual se consideran redes RNR con más de una capa oculta o multicapa, aunque puede ser empleada en redes de una sola capa. Si se habla de la reducción del error cuadrático medio global, se supone que la superficie de la función error, es una función cuadrática. Para una RNR multicapa la superficie del error no es una función cuadrática; su forma es diferente para diferentes regiones del espacio, y la velocidad de convergencia puede incrementarse por la variación de la Razón de Aprendizaje, en cada región diferenciada de la superficie del error, siempre que no se sobrepase cierto valor máximo pre-definido, para lograr un aprendizaje estable.

De modo que *entrenar la RNR por este método, supone un riesgo aún menor, de arrojar una convergencia prematura, “estancándose” en un mínimo local de la función error cuadrático medio global.*

Sin embargo, aquí intervienen algunas consideraciones para el diseño de la RNR solidaria a la Hiperheurística propuesta:

**a)** cualquiera de estos métodos, supone un problema de optimización no lineal, que, como es sabido, no proporciona, por medio de métodos clásicos, una garantía de convergencia hacia un mínimo minimorum error cuadrático medio global;

**b)** La capacidad predictiva de la RNR, es decir, que no falle en los patrones de salida cuando el patrón de estímulo en la entrada fue observado en su entrenamiento, y que logre resolver patrones de salida ante estímulos de entrada no observados durante su entrenamiento, depende de dos factores interdependientes, excluyendo el diseño (que es un problema en sí mismo): 1ro) del número de patrones de entrada y su correspondientes patrones de salida, empleados durante el entrenamiento de la RNR. A mayor número, mayor capacidad predictiva; y 2do) de que el algoritmo de entrenamiento converja en el mínimo minimorum error cuadrático medio global o próximo a él. A menor error, mayor capacidad predictiva.

**c)** En este punto es considerado el diseño de la RNR que soporta a la HY X-FPSO CBR. Se observa que se ha optado por emplear patrones de entrada y salida binarios, no números reales. Hay dos razones que sustentan esta elección, ambas compatibles con: el dominio compuesto por las 4 Metaheurísticas X-FPSO, su habilidad para satisfacer los 5 Principios de IG, y las características del Espacio de Soluciones, mencionadas en los desarrollos teóricos de la Hiperheurística propuesta. La primera: resulta mucho más sencillo decodificar e interpretar la información obtenida desde los patrones de salida binarios de la RNR, al momento de seleccionar la X-FPSO forma más apta para continuar la estrategia de búsqueda en el espacio de soluciones. Asimismo, resulta mucho más simple codificar la información en los patrones de entrada. La segunda: el número posible de patrones de entrada y de salida, está acotado por las variaciones con repetición de 2 elementos (0 y 1) tomados de  $n_{NE}$ , para la los patrones de entrada, y de  $n_{NS}$ , para los patrones de salida, respectivamente. Es decir: si  $n_{PEMax}$  y  $n_{PSMax}$  son los máximos números de patrones de entrada y salida, correspondientemente, se tendrá:

$$n_{PEMax} = 2^{n_{NE}} \quad (7)$$

$$n_{PSMax} = 2^{n_{NS}} \quad (8)$$

Para el diseño propuesto, la cota en el conjunto de patrones de entrada resulta  $n_{PEMax} = 2^{15} = 32768$  patrones, y la cota en el conjunto de patrones de salida resulta  $n_{PSMax} = 2^7 = 128$  patrones. Se infieren dos características: a) una gran cantidad de patrones de entrada posibles en el conjunto  $[1 \dots n_{PEMax}]$ , no tienen sentido práctico y b) muchos de ellos pueden conducir al mismo patrón de salida en el conjunto  $[1 \dots n_{PSMax}]$ . En menor grado, puede que algunos patrones de salida sean redundantes o no requeridos. Pero, sin duda, con este diseño, se tiene una cota en los conjuntos de patrones entrada y salida, y un método para definir los mismos, situación que en otros diseños, en extremo dificultosa o imposible. Aquí se tiene un aporte de eficiencia en el diseño de la RNR, conforme las características del diseño de la HY X-FPSO CBR, provenientes de las Metaheurísticas X-FPSO seleccionadas, y de la Clase de Problemas a abordar, los cuales definen un particular Espacio de Soluciones.

**d)** De modo que el aspecto diseño de la RNR, puede asumirse eficiente. Pero queda aún latente el riesgo de pérdida de capacidad predictiva, que en esta aplicación es crucial, si se emplean métodos clásicos para la minimización del error cuadrático medio global referido. Estos métodos pueden funcionar muy adecuadamente en otros contextos de aplicación. Pero, luego de múltiples simulaciones, no fue este diseño el caso.

**e)** Se propone, entonces, una solución referida como MatHeurística, otro aporte parcial del presente trabajo: Se hibridiza el Algoritmo de Entrenamiento de la RNR, aplicando, en iteraciones tempranas, el Backpropagation con Gradiente Descendente y Momento. Luego, en un punto de corte convenientemente definido (por caso, se alcanza cierto error parcial mínimo), y conteniendo las Matrices de Pesos Sinápticos y Biases, **[WOE]**, **[BO]**, **[WSO]** y **[BE]**, valores pertinentes en la estrategia de minimización, es

aplicada una Metaheurística PSO (Mono-Objetivo) con Factor de Constricción (forma X2-PSO). Los resultados obtenidos fueron muy favorables.

### 3. APLICACIÓN DE LA HY X-FPSO CBR PARA LA IDENTIFICACIÓN DEL ESPACIO DE ESTADOS REQUERIDO EN LA OPTIMIZACIÓN DINÁMICA POSIBILÍSTICA.

#### 3.1. Los objetivos de optimización para la definición del espacio de estados requerido por la optimización dinámica posibilística.

A los efectos de aplicar la Hiperheurística propuesta, deben identificarse los objetivos de optimización, *critérios de mérito en la evolución del SDEE* en el Período de Control Regulatorio. A partir de los mismos, se procederá a describir el método que conduce a la definición del Espacio de Estados, para un número de etapas  $N=5$ , pues tal período es quinquenal en Argentina, y el SDEE considerado en la simulación pertenece a Bariloche, provincia de Río Negro. El SDEE, su Período de Control Regulatorio y los objetivos (*critérios de mérito que deben ser minimizados*), siguen la referencia [Schweickardt & Miranda, 2009], a los efectos de comparar el Espacio de Estados obtenido, con el que arroja la HY X-PSO CBR. Los criterios son:

{1-Costo Global (CG); 2-Energía No Suministrada (ENS); 3-Índice de Interrupción del Servicio (FI); 4-Índice de Tensión Fuera de Tolerancia (TF); 5-Pérdidas Globales de Potencia (PG); 6-Índice de No-Calidad (Impacto) Ambiental (Visual) por empleo de Típicos Constructivos de Líneas fuera del Establecido según Zona (IALin); 7- Índice de No-Calidad (Impacto) Ambiental (Visual) por empleo de Típicos Constructivos de Centros de Transformación fuera del Establecido según Zona (IACT); 8-Flexibilidad del SDEE (FLEX) }

Por su complejidad, no se expondrán aquí desarrollos, tales como los que conducen a los Índices, ya que pueden ser consultados en la citada referencia y no agregan valor a la presentación de este trabajo. Si al acrónimo de cada criterio de mérito, se le antepone la letra  $v$ , se tendrá el siguiente *vector de variables asociadas a los objetivos*:  $\{CvCrit\} = \{vCG; vENS; vFI; vTF; vPG; vIALIN; vIACT; vFlex\}$ . Para los Índices, coinciden **Índice** y *variable*. La Demanda a satisfacer en cada año de corte, *se supone determinística* (no tiene por qué serlo y podría ser extendida como *restricción difusa*, al igual que las variables solidarias a los criterios) y *proveniente de un modelo prospectivo*.

Adicionalmente, *se requiere de una extensión al dominio difuso de tales variables*, para implementar las X-FPSO formas. La expresión genérica propuesta a tal fin, resultó desde: si  $m$  es la variable genérica, en el conjunto  $\{CvCrit\}$ , entonces:

$$\mu_m = 1; \text{ si } vMin_m \geq vm \quad (9)$$

$$\mu_m = \left( \frac{(vMax_m - vm)}{vMax_m - vMin_m} \right)^{p_\mu^m} \quad (10)$$

; si  $vMin_m \leq vm \leq vMax_m$

$$\mu_m = 0; \text{ si } vMax_m \leq vm \quad (11)$$

Siendo:

Además  $p_\mu^m$  el *ponderador exponencial (contractor/dilatador)* correspondiente a cada Conjunto Difuso, y  $vMax_m$ ,  $vMin_m$  los valores *máximo* y *mínimo* límites, respectivamente, para cada variable  $m$  en  $[1..M]$ , con  $M=8$ , en este caso.

La *decisión maximizante estática* que mide la *aptitud* en el Conjunto Difuso de Decisión, viene dada, como se presentó en el modelo teórico de la Hiperheurística propuesta, por la *t-norma* Producto de Einstein [Schweickardt & Pistonesi, 2010]:

$$fapD = \mu_D^{Max} = t_{PE}^{Max} \{ \mu_{m1}; \mu_{m2}; \dots; \mu_{mM} \} \quad (12)$$

### 3.2. La Aplicación de la Hiperheurística HY X-FPSO CBR

#### 3.2.1. Método de Aplicación conforme los Criterios de Optimización del SDEE y sus Variables de Decisión en el Espacio de Soluciones

La Optimización Dinámica Posibilística asociada a la Planificación del SDEE, se acota al Subsistema de Media Tensión, en 13,2 [kV]. Se parte de un estado de referencia para el SDEE, coincidente con el comienzo del quinquenio de Control Regulatorio, y único estado de la Etapa 0. En el mismo, se relevan los valores correspondientes a cada variable  $m$  solidaria a los criterios de mérito del sistema. El método propuesto para generar el conjunto de estados que le corresponde a cada etapa, consiste en definir un número  $PO$  de Matrices de Preferencias entre dichos criterios, las cuales arrojarán Vectores de Prioridades Exponenciales, cuyas componentes serán los valores  $p_\mu^m$  ( $po$ ), con  $po$  en  $[1..PO]$ . La dimensión del espacio de soluciones, estará definida por el número de variables de control o decisión en el problema Multi-Objetivo,  $nvD$ . En esta aplicación, para la planificación de SDEE, se tiene, si  $vDd$  es la variable de decisión genérica con  $d$  en  $[1..nvD]$ :

$[vDd]$ : **vD1**: *Nro de Alimentadores*; **vD2**: *Ubicación o Emplazamiento de los Alimentadores* (se dispone de cierto número de tendidos posibles, resultado de la Planificación de Largo Plazo); **vD3**: *Tipo de Alimentador* (Aéreo Convencional, Aéreo Preensamblado, Subterráneo); **vD4**: *Sección del Alimentador* (están normalizadas en: 120, 95, 70, 50, 35 y 25 [mm<sup>2</sup>]); **vD5**: *Tipo de Centro de Transformación (CT)* (Plataforma, A

Nivel, Subterráneo); **vD6**: *Ubicación o Emplazamiento de los CT* (también se dispone de un número de posibles ubicaciones de los CT, resultado de la Planificación de Largo Plazo); **vD7**: *Potencia de los CT* (normalizadas en 50, 100, 150, 200, 250, 300, 500, 1000 [kVA]); **vD8**: *Número de Dispositivos de Conmutación (Seccionadores)* para transferencia de carga; **vD9**: *Ubicación de los Dispositivos de Conmutación* (también se tiene un número de posibles ubicaciones, resultado de la Planificación de Largo Plazo); **vD10**: *Potencia en Generación de Reserva en cada Estación Transformadora 33/13.2 [kV]* (4; 3 existentes + 1 resultado de la Planificación de Largo Plazo. Tales Potencias se discretizan en el conjunto [5, 10, 15, 20, 25] [MW]). Por tanto **nvD = 10**, con algunos de sus dominios, discretos, ya establecidos.

De este modo, en ese espacio de soluciones se aplicarán las formas X-FPSO, controladas por la Hiperheurística propuesta, conforme su RNR. El procedimiento se sintetiza en los siguientes pasos:

**1ro)** Determinar el Número PO de Matrices de Preferencia entre los Criterios definidos para el mérito del SDEE;

**2do)** Obtener el Vector de Prioridades para cada una de ellas;

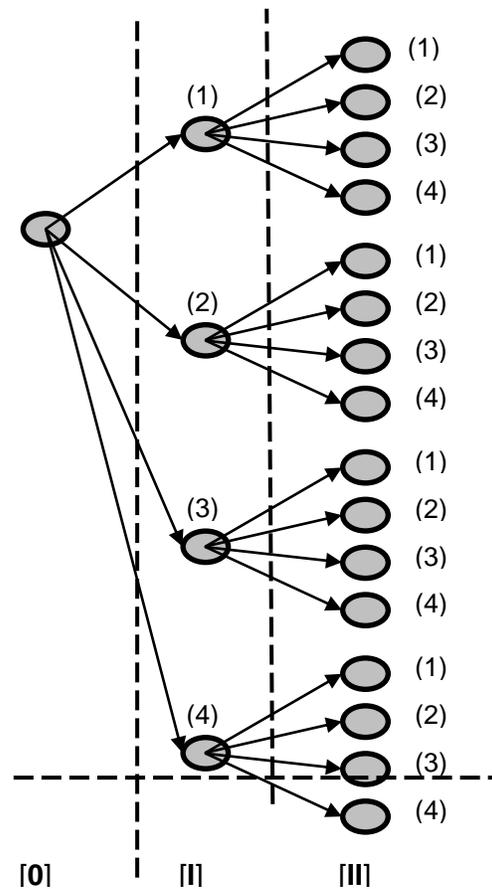
**3ro)** Iterar desde  $k=1$  hasta  $k=N$ ,  $N$ =Número de Etapas de la Optimización Dinámica Posibilística;

**A)** Iterar desde  $i=0$  (estado factible de referencia) hasta el número de estados factibles determinados en la etapa  $k$ ;

**A-1)** Iterar, desde  $p_0 = 1$  hasta PO haciendo: **a)** la Construcción de los Conjuntos Difusos, según la ponderación  $p_{\mu}^m(p_0)$ , desde las expresiones (9), (10) y (11),

**B)** Aplicación de la HY X-FPSO CBR, como transición entre estado  $i$  y el que se definirá, con la restricción determinística de demanda para la etapa  $k$ , obteniendo la solución más satisfactoria. La misma resulta de la t-norma Producto de Einstein, aplicada entre las funciones de pertenencia solidarias a cada Conjunto Difuso, según la expresión (12). Se logra así un estado factible del Espacio de Estados en la etapa  $k$ ; al ir incrementando  $p_0$ , y luego  $I$ , se obtienen todos los estados factibles de la etapa  $k$ ; al ir incrementando  $k$ , y repitiendo los pasos de iteración, se obtiene el Espacio de Estados completo. Si  $PO = 4$ , tal como se consideró en la simulación, la construcción del Espacio de Estados para las Etapas [0], [I] y [II], se observa en la Figura 5. Cada estado, se obtiene, por aplicación de la HY F-PSO CBR según el valor de prioridad  $p_{\mu}^m(p_0)$ , mediante las iteraciones anteriores.

Los  $p_{\mu}^m(p_0)$ , surgen de PO Matrices de Preferencia Determinísticas entre los criterios de optimización. Sin embargo, en este Modelo se propone incorporar



**Figura 5:** Esquema de la Determinación del Espacio de Estados para las Etapas [0], [I] y [II], con  $PO = 4$

**Fuente.** Elaboración propia.

incertidumbres de valor en las mismas, y representarlas mediante Números Difusos, tal y como se desarrolla en [Schweickardt & Miranda, 2009]. A su vez, la obtención del Vector de Prioridades, no sigue el modelo Autovalor/Autovector de Saaty [Saaty, 1977], ni el complejo modelo, más exacto, propuesto en la referencia anterior. Sigue un Modelo Híbrido: Calcula las Matrices de Preferencias Determinísticas como se explica en esa referencia, y se desarrolla en el epígrafe siguiente. Luego, sobre esa Matriz Determinística, calcula el autovector de Perron [Lax, 1992], y desde allí se obtiene el Vector de Ponderadores Exponenciales.

### 3.2.2. Obtención de los PO Vectores de Prioridades desde Matrices de Preferencias Difusas

Si, **pref** es un Número Difuso (ND), expresado como:

$$\forall \alpha \in [0,1], \text{pref} = [ \text{pref}_1(\alpha), \text{pref}_2(\alpha) ] \quad (13)$$

Entonces se tendrá una Matriz de Preferencias entre Criterios Difusa:

**MPA:**  $\forall \alpha \in [0,1] \rightarrow$

$$\begin{pmatrix} 1 & [pref_1(\alpha), pref_2(\alpha)]_{12} & \dots & [pref_1(\alpha), pref_2(\alpha)]_{1n} \\ [1/pref_2(\alpha), 1/pref_1(\alpha)]_{21} & 1 & \dots & [pref_1(\alpha), pref_2(\alpha)]_{2n} \\ \dots & \dots & \ddots & \dots \\ [1/pref_2(\alpha), 1/pref_1(\alpha)]_{n1} & [1/pref_2(\alpha), 1/pref_1(\alpha)]_{n2} & \dots & 1 \end{pmatrix} \quad (14)$$

Las incertidumbres de cualquier preferencia **prefij** y de su recíproca **prefji**, son dependientes. Esto significa que si se presentase una ocurrencia de **prefij** en el segmento de confianza limitado por  $\alpha$ , entonces:  $pref_{ji}(\alpha) = 1/ pref_{ij}(\alpha)$ . Con ello se garantiza que cualesquiera sean las ocurrencias en sus entradas, la Matriz de Preferencias **MPA**( $\alpha$ ) es determinística, puesto que es una instancia ( $\alpha$ ) de **MPA**. Adicionalmente, siempre será positiva y recíproca. **MPA**( $\alpha$ ) será referida como Matriz de Preferencias Colapsadas según el Nivel de Certidumbre  $\alpha$ . Las PO instancias distintas de esta **MPA**( $\alpha$ ), arrojarán los PO Vectores de Prioridad Exponenciales necesarios. Esto se logra modificando PO veces el nivel de certidumbre  $\alpha$ . Si las **prefij** son representadas por ND Triangulares (NDT), y L, R son sus funciones a Izquierda y Derecha, respectivamente, con prefMP el valor central o de máxima posibilidad, el Criterio de Colapso Removal, *Rv*, empleado, se define como:

$$Rv[pref(ac)] = pref_{MP} + \frac{1}{2} \times [IDer - Ilzq] \quad (15)$$

$$IDer = \int_{pref_{MP}(ac)}^{pref_2} R(pref) dpref \quad (16)$$

$$Ilzq = \int_{pref_1(ac)}^{pref_{MP}} L(pref) dpref \quad (17)$$

Pero si el valor representativo de las preferencias difusas para cierto ( $ac$ ), está dado únicamente por (15)-(17), no se estaría considerando la consistencia entre las mismas, conforme la expresión:

$$pref_{ik}(ac) \times pref_{ij}(ac) = pref_{jk}(ac) \quad (18)$$

Dentro del segmento de confianza fijado por ( $ac$ ), se requiere la búsqueda de aquellos valores tales que la matriz **MPA**( $ac$ ) resulte lo más consistente posible. De modo que los valores representantes de las preferencias dentro del segmento ( $\alpha = ac$ ), tendrán que satisfacer dos objetivos: **1**) que se aparten lo menos posible de su *Rv*( $ac$ ) y **2**) que satisfagan lo más posible las ecuaciones de consistencia. El Sistema de Ecuaciones de Consistencia, respetando la formulación (18), para  $n$  criterios (orden de la matriz  $n \times n$ ), se expresa del modo siguiente:  $\forall \alpha \in [0,1]$

$$Sea C = \{ \forall i \in [2..n-1]; \forall j \in [i+1.. n] y \forall k \in [1.. i-1] \}$$

entonces:

$$\{ pref_{ij}(ac) = pref_{kj}(ac)/pref_{ki}(ac) \} \quad (19)$$

Los dos objetivos según **1**) y **2**) en el punto anterior, pueden ser planteados en un Programa Lineal. Para ello, los errores ( $e$ ) entre cada preferencia  $pref_{ij}(ac)$  y su *Rv*[ $pref_{ij}(ac)$ ], y entre cada preferencia  $pref_{ij}(ac)$  y su formulación consistente según (19), pueden introducirse como factores. Por caso, si se buscara la máxima consistencia en cierta ecuación de sistema (19) en el conjunto  $C$ , se tendría:

$$pref_{ij}(ac) \times ec_{ij}^k = pref_{kj}(ac)/pref_{ki}(ac) \quad (20)$$

si  $ec_{ijk} = 1$ , entonces la consistencia resultaría perfecta. Puede observarse que  $0 < ec_{ijk} \leq 1$ . De igual modo, para el caso del apartamiento mínimo de  $pref_{ij}(ac)$  respecto del *Rv*[ $pref_{ij}(ac)$ ], puede formularse la expresión:

$$pref_{ij}(ac) \times erv_{ij} = Rv[pref_{ij}(ac)] \quad (21)$$

con  $0 < erv_{ij} \leq 1$ . En consecuencia, el modelo puede linealizarse en sus restricciones, empleando variables logarítmicas. Sus objetivos serían la minimización, respectivamente, de la sumatoria de los valores absolutos de los logaritmos de los errores,  $Lec_{ij}^k$ ,  $ALec_{ij}^k$  y  $erv_{ij}$ ,  $ALerv_{ij}$ . En principio, la introducción de la operación valor absoluto (considerando que pueden existir errores logarítmicos menores que cero), parecería generar objetivos no lineales. Esta cuestión se resuelve con el agregado de restricciones de desigualdad, que relacionen las variables asociadas a los valores absolutos de los errores logarítmicos, con los errores logarítmicos. Ambos objetivos, pueden ponderarse creándose una única función a minimizar: el valor absoluto del error total ponderado,  $ALerrT$ , conforme los ponderadores  $P_c + P_{rv} = 1$ . De modo que, bajo estas consideraciones, el problema de *optimización lineal* que resuelve el conjunto de preferencias más representativo en la matriz **MPA**( $ac$ ), se formula como sigue:

$$\begin{aligned} &Min \\ &\left\{ ALerrT = \rho_c \times \left( \sum_{i=2}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^{i-1} ALec_{ij}^k + \rho_{rv} \times \left( \sum_{i=1}^{n-1} \sum_{j=i+1}^n ALrv_{ij} \right) \right) \right\} \quad (22) \end{aligned}$$

Sujeto a:  
[Restricciones de Consistencia]

Sea  $C = \{ \forall i \in [2..n-1]; \forall j \in [i+1.. n] y \forall k \in [1.. i-1] \}$ , entonces:

$$Lpref_{ij}(ac) + Lec_{ij}^k = Lpref_{kj}(ac) - Lpref_{ki}(ac), \text{ en } C \quad (23)$$

[Restricciones de VA de los Errores Logarítmicos de Consistencia]

$$ALec_{ijk} - Lec_{ij}^k \geq 0, \text{ en } C \quad (24)$$

$$ALec_{ijk} + Lec_{ij}^k \geq 0, \text{ en } C \quad (25)$$

$$ALec_{ijk} \geq 0, \text{ en } C \quad (26)$$

[Restricciones Apartamiento respecto de  $Rv[pref_{ij}(ac)]$   
Sea  $CI = \{ \forall i \in [1..n-1]; \forall j \in [i+1.. n] \}$ , entonces:

$$Lpref_{ij}(ac) + Lerv_{ij} = LRv[pref_{ij}(ac)], \text{ en } CI \quad (27)$$

[Restricciones de VA de los Errores Logarítmicos de  $Rv[pref_{ij}(ac)]$

$$ALerv_{ij} - Lerv_{ij} \geq 0, \text{ en } CI \quad (28)$$

$$ALerv_{ij} + Lerv_{ij} \geq 0, \text{ en } CI \quad (29)$$

$$ALerv_{ij} \geq 0, \text{ en } CI \quad (30)$$

[Restricciones de Segmento de Confianza [1,2] al Nivel  $(ac)$

$$Lpref_{ij}(ac) \geq Lpref_{ij}(ac)_1, \text{ en } CI \quad (31)$$

$$Lpref_{ij}(ac) \leq Lpref_{ij}(ac)_2, \text{ en } CI \quad (32)$$

Siendo:

$\rho_c$  y  $\rho_{rv}$  los ponderadores fijados para los objetivos ( $\rho_c + \rho_{rv} = 1$ );  $Lec_{ij}^k$  el logaritmo (en base  $e$ , por caso) del error multiplicativo  $ec_{ij}^k$  y  $ALec_{ij}^k$  su valor absoluto;  $Lerv_{ij}$  el logaritmo del error multiplicativo  $erv_{ij}$  y  $ALerv_{ij}$  su valor absoluto;  $Lpref_{ij}(ac)$  el logaritmo del valor de la preferencia  $pref_{ij}(ac)$ ;  $[Lpref_{ij}(ac)_1; Lpref_{ij}(ac)_2]$  el Segmento de Confianza logarítmico al nivel de certidumbre  $(ac)$  ( $ac$  es dato para el modelo);  $LRv[pref_{ij}(ac)]$  es el logaritmo del colapso Removal aplicado sobre  $pref_{ij}(ac)$ ;  $ALerrT$  es el error logarítmico ponderado total, por inconsistencias y por apartamientos respecto a los correspondientes colapsos  $Rv[pref_{ij}(ac)]$ . Resuelto este programa lineal, las preferencias son obtenidas por exponenciación de los valores logarítmicos según la base considerada. Si la base es el número  $e$ :

$$Pref_{ij}(ac) = e^{Lpref_{ij}(ac)}, \text{ en } CI \quad (33)$$

Resultando valores que no necesariamente son enteros en  $[1..10]$ . Tal especificación de escala, propuesta por Saaty, se torna carente de sentido al formular una solución de preferencias difusas colapsadas, de mínima inconsistencia. Para tales valores de  $pref_{ij}(ac)$ , que componen la  $MPA(ac)$  determinística, se calcula el autovector de Perron y desde allí, normalizándolo, componentes  $vp_i^{[N]}(ac)$ , y luego multiplicando sus componentes por el número  $n$  de criterios, el Vector de Ponderadores Exponenciales buscado, componentes  $vp_i^{[N]}(ac) \equiv p_\mu^m$  (po) para cierto po- $ac$ -corte, definido por po en  $[1..PO]$ :

$$vp_i^{[N]}(ac) = \frac{vp_i(ac)}{\sum_{i=1}^n vp_i(ac)}, \text{ en } CI \quad (34)$$

$$vp_i^{[E]}(ac) = n \times vp_i^{[N]}(ac), \text{ en } CI \quad (35)$$

#### 4. SIMULACIÓN SOBRE UN SDEE REAL

La simulación del presente modelo, se ha realizado sobre un SDEE real, emplazado en la Patagonia Argentina, en la ciudad de Bariloche, provincia de Río Negro. El análisis se corresponde con el período de control tarifario 2003-2007. El SDEE de Bariloche cubre un área de 350 [km<sup>2</sup>]. Sirve a unos 40000 usuarios. Es abastecido en 33 [kV] (Subtransmisión - SbT) y tiene 3 existentes + 1 proyectada en el período, Estaciones Transformadoras 33/13.2 [kV] (Media Tensión - MT). El sistema de Media Tensión, tiene cerca de 500 Centros de Transformación de 13.2/0.38 [kV] (Baja Tensión - BT). Como era de esperar, el Espacio de Estados (EE) quedó conformado por la misma cantidad de estados (variantes de solución/equipamiento) que en la referencia [Schweickardt & Miranda, 2009]. De modo que el Problema de Optimización Dinámica Posibilística se divide en 5 Etapas: {**Etapa I:** Referencia **Etapa II: 5 estados; Etapa III: 4 estados; Etapa IV: 4 estados; Etapa V: 3 estados; Etapa VI: 1 estado** (final)}. En la **TABLA 1**, se presenta la Trayectoria Más Satisfactoria, TMS, para un  $ac = 0.25$ . El Vector de Ponderadores Exponenciales empleado, se indica en la segunda fila.  $\Theta_{Ext} = 0.35$  es el Riesgo Extrínseco fijado, y debe ser mayor o igual que  $1 - \mu D(TMS)$ . La **TABLA 2** presenta los 4 Vectores de Ponderación Exponencial empleados para generar el EE. En la **TABLA 3** se presentan las Transiciones de la Optimización Posibilística, soportada en Programación Dinámica Difusa, en el Espacio de Estados obtenido por la HY X-FPSO CBR.

#### 5. CONCLUSIONES

Se han presentado los conceptos específicos para la aplicación de una novedosa Hiperheurística basada en Razonamiento, mediante una Red Neuronal Tipo Retropropagación, y soportada en cuatro formas X-FPSO, como dominio. Es referida como HY X-FPSO CBR. Los conceptos más novedosos, fueron establecidos a lo largo del presente trabajo, complementando su parte teórica. Dos conclusiones fundamentales, se relacionan con su aplicación: **1ra)** Es claro que un problema de las características planteadas (*establecer el Espacio de Estados en una Optimización Dinámica Posibilística*), no podría ser resuelto sino por una metodología de este tipo y **2da)** Se han asumido técnicas y métodos auxiliares, como estrategia de variación de ponderadores exponenciales, iterando la Hiperheurística sobre ellos. Esto permite observar el vastísimo campo de investigación solidario a estas ideas, extendiéndose más allá del ámbito de los SDEE, trabajado, por su conocimiento, por los autores. La Hiperheurística propuesta, se plantea como una interesante estrategia para toda Clase de Problemas, que reúnan las condiciones generales descritas.

pCGE	pENS	pFI	pTF	pPG	pIALin	pIACT	pFLX		
1.97	2.01	0.75	0.79	0.95	0.49	0.49	0.55		
E, e	μD(TMS)	μCG	μENS	μFI	μTF	μPG	μIALin	μIACT	μFLX
I, 1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
II, 3	0.82	1.00	1.00	0.82	0.87	1.00	0.94	0.87	1.00
III, 2	0.82	0.96	1.00	0.88	0.86	1.00	0.91	0.99	0.99
IV, 3	0.82	1.00	1.00	1.00	0.83	1.00	0.93	0.91	1.00
V, 1	0.78	0.89	0.78	1.00	0.85	0.98	0.82	0.88	0.98
VI, 1	0.78	1.00	1.00	0.97	0.85	0.98	0.89	0.80	1.00

**Tabla 1:** Cálculo de la TMS(ac):  $ac = 0.25$  y  $\Theta_{Ext} = 0.35$

Fuente. Elaboración propia

pCGE	pENS	pFI	pTF	pPG	pIALin	pIACT	pFLX
1.97	2.01	0.75	0.79	0.95	0.49	0.49	0.55
2.00	1.87	0.63	0.81	0.43	0.89	0.89	0.48
1.25	2.72	0.71	0.87	0.92	0.57	0.57	1.39
1.12	1.92	0.83	0.65	1.15	0.55	0.58	1.17

**Tabla 2:** Vectores de Ponderación Exponencial resultantes de las Matrices de Preferencias Colapsadas en 4 ac-cortes, empleados para la generación del Espacio de Estados. Por condición:  $\Sigma p = \delta =$  número de Criterios.

Fuente. Elaboración propia

[Ei,ei; Ef, ef]	CG	ENS	FI	TF	PG	IALin	IACT	FLEX
[I, 1; II, 1]	188.0	17050	0.73	0.045	185.40	0.719	0.558	169.2
[I, 1; II, 2]	193.3	7930	1.19	0.048	210.45	0.983	1.072	173.9
[I, 1; II, 3]	186.8	6919	1.02	0.046	150.04	0.325	0.981	168.1
[I, 1; II, 4]	177.7	14994	0.81	0.040	178.90	0.067	0.149	159.9
[I, 1; II, 5]	189.3	9144	0.50	0.039	190.65	0.299	0.542	170.4
[II, 1; III, 1]	137.8	9042	0.89	0.047	150.00	0.841	0.358	124.1
[II, 1; III, 2]	141.4	11656	0.98	0.039	178.65	0.426	0.155	127.3
[II, 1; III, 3]	149.1	17224	1.02	0.039	145.66	0.064	0.287	134.2
[II, 1; III, 4]	230.7	10196	0.89	0.038	201.54	0.633	0.073	207.6
[II, 2; III, 1]	141.3	13443	1.46	0.042	186.54	0.517	0.102	127.2
[II, 2; III, 2]	217.2	13117	0.64	0.040	170.65	0.634	0.210	195.5
[II, 2; III, 3]	150.3	9708	1.65	0.046	164.30	1.516	1.365	135.2
[II, 2; III, 4]	328.7	9083	0.66	0.040	180.43	0.467	0.429	295.9
[II, 3; III, 1]	172.3	17377	1.43	0.041	193.45	1.356	0.918	155.1
[II, 3; III, 2]	124.4	10094	0.94	0.044	158.69	0.454	0.059	111.9

**Tabla 3:** Espacio de Estados obtenido y Transiciones [Etapa, estado], i: inicial, f: final, posibles, desde la aplicación de la Optimización Dinámica Posibilística. Valores de las variables asociadas a cada Criterio

Fuente. Elaboración propia

## REFERENCIAS

Funahashi K., Nakamura Y. (1993): "Approximation of Dynamical System by Continuous Time Recurrent Neural Networks". Journal of Neural Networks, Nro. 6, pp. 801-806.

Hagan M., Demuth H., Beale M. (1996): "Neural Network Design". PWS Publishing Company. Boston, USA.

Hilera R., Martínez J. (1995): "Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones". Ra-ma Editorial. Madrid, España.

Lax P. (1997): "Linear Algebra". Wiley Interscience: New York

Saaty T. (1977): "A Scaling Method for Priorities in Hierarchical Structures". Journal of Mathematical Psychology, 15, 234-281.

Schweickardt G., Miranda V. (2009): "A Two Stage Planning and Control Model Toward Economically Adapted Power Distribution Systems Using Analytical Hierarchy Processes and Fuzzy Optimization". International Journal of Electrical Power and Energy Systems, Vol 2., issue 4. Pp. 256-274. Elsevier.

Schweickardt G., Pistonesi H. (2010): "Un Modelo Posibilístico para estimar el Costo Intrínseco de la Energía No Suministrada". Revista DYNA, Nro. 162, pp. 249-259. Universidad Nacional Colombia.

Simpson P. (1992): "Foundations of Neural Networks". Artificial Neural Networks. IEEE Press. New York.

Widrow B., Lehr M., (1990): "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation". Proceedings of the IEEE, Vol. 78 issue 9.