

HIPERHEURÍSTICA BASADA EN RAZONAMIENTO CON DOMINIO EN METAHEURÍSTICAS X-PSO MULTIOBJETIVO HY X-FPSO CBR.

APLICACIÓN SOBRE UNA OPTIMIZACIÓN DINÁMICA POSIBILÍSTICA PARTE 1) DESARROLLOS TEÓRICOS DEL ALGORITMO HIPERHEURÍSTICO HY X-FPSO CBR

GUSTAVO SCHWEICKARDT - CARLOS CASANOVA - JUAN MANUEL GIMENEZ
CONICET - Universidad Tecnológica Nacional, Concepción del Uruguay - ARGENTINA
CONICET - Universidad Nacional de San Juan - Facultad de Ingeniería
San Juan - ARGENTINA

schweickardt@frcu.utn.edu.ar - casanovac@frcu.utn.edu.ar - jgimenez@uns.edu.ar

Fecha Recepción: Agosto 2013 - Fecha Aceptación: Octubre 2013

RESUMEN

En el presente trabajo se desarrolla el marco conceptual/teórico relativo a una novedosa HiperHeurística, basada en Razonamiento y aplicada en el dominio de MetaHeurísticas variantes de la Optimización Por Enjambre de Partículas (PSO), denominadas X-PSO, MultiObjetivo. Esta HiperHeurística, referida como HY X-FPSO CBR (*Case Based Reasoning*) emplea, como mecanismo de selección de la forma X de la MetaHeurística FPSO a ser aplicada en cierta instancia de decisión, una Función de Elección aproximada mediante una Red Neuronal Artificial tipo Retropropagación. Para el diseño y, particularmente, entrenamiento de la misma, son considerados aspectos relativos a los Principios de la Inteligencia de Grupo y las habilidades que cada forma X-FPSO exhibe para satisfacerlos, así como las características del Espacio de Búsqueda, inherentes a la Clase de Problemas que deben resolverse mediante la HiperHeurística propuesta: Establecer el Espacio de Estados requerido por una Optimización Dinámica Posibilística sobre la Planificación de Mediano/Corto Plazo de un Sistema de Distribución de Energía Eléctrica (SDEE).

PALABRAS CLAVE: Optimización – Enjambre de Partículas – HiperHeurísticas
Redes Neuronales Artificiales – Sistema de Distribución de Energía Eléctrica

ABSTRACT

In this work the conceptual/theoretical framework of a novel HyperHeuristic, Case Based Reasoning and supported on some variants of MultiObjective Particle Swarm Optimization MetaHeuristic, called X-PSO, are presented.

This HyperHeuristic, referred as HY X-FPSO CBR (Case Based Reasoning), works by mean of selection function, aproximated with an Artificial Backporpagation Neural Network. To design and, especially, training the Artificial Neural Network, Swarm Intelligence Principles and the skill that each X-FPSO form exhibit to satisfy it, as well as the Search Space that is defined by the Problems Class to solve, are considered. This HyperHeuristic are designed to be applied in the definition of States Space required for a Possibilistic Optimization on the Mid/Short term Planning of a Electric Distribution System (EDS).

KEYWORDS: Optimization – Particles Swarm – HyperHeuristics – Artificial Neural Network – Electric Distribution System

1. INTRODUCCIÓN

La aplicación de HiperHeurísticas en problemas para los que se pretenden soluciones de calidad, obtenidas en un tiempo menor al que proporcionan los Algoritmos Heurísticos o MetaHeurísticos, tiene una trayectoria de larga data en el estado del arte. Se podrá apreciar, en la Sección 2., la extensa bibliografía analizada al respecto, y los años en los que sus aportes han sido introducidos. Sin embargo, en algún punto de los desarrollos de la investigación, comenzaron, de manera paralela, a buscarse mejoras, por un lado, en la autoadaptación de los parámetros de diversas MetaHeurísticas, extensiones de las mismas al dominio MultiObjetivo y a la solución de Clases de Problemas, en lugar de obtener especializaciones sobre problemas específicos, y, por el otro, a proponer una línea de evolución de las HiperHeurísticas que sea capaz de realizar las mismas tareas. Si bien se presentará una definición formal, la diferencia entre una HiperHeurística y una MetaHeurística estriba en el mayor nivel de abstracción que la primera exhibe respecto de la segunda: la HiperHeurística tiene como dominio el espacio de Heurísticas o MetaHeurísticas, mientras que la MetaHeurística se aplica sobre el espacio de soluciones. Parece razonable componer una estrategia que utilice las mejores Heurísticas o MetaHeurísticas, en cierta instancia de decisión, para continuar la búsqueda de soluciones, avanzando en una mejora de la calidad de las mismas y minimizando el tiempo requerido. En este trabajo es propuesta una HiperHeurística novedosa, que aprovecha los desarrollos MetaHeurísticos en torno a la Inteligencia de Grupo y la caracterización del Espacio de Búsqueda que los problemas de Optimización, en el dominio de los Sistemas de Distribución Eléctrica (SDEE), exhiben. Puede hablarse por ello de una Clase de Problemas, pero sin acotarlos a los SDEE. En esta primera parte del trabajo, se establecen los lineamientos teóricos para el diseño de la aquí referida como HiperHeurística Basada en Razonamiento (*Case Based Reasoning* - CBR) con Dominio en Variantes X de la MetaHeurística PSO MultiObjetivo: HY X-FPSO CBR.

Como se verá, la extensión de una forma X-PSO MonoObjetivo, al dominio MultiObjetivo, se sustenta en Conjuntos Difusos (*Fuzzy Sets*). De aquí el acrónimo X-FPSO, en lugar de referir X-PSO MultiObjetivo.

El trabajo se organiza como sigue: En la Sección 2., se presentan los conceptos de Heurística, MetaHeurística e HiperHeurística, con detenimiento en esta última, particularmente en la clasificación relativa a los métodos reconocidos, en el estado del arte, para el Aprendizaje/Selección de la Heurística/MetaHeurística más apta a aplicar en cierta instancia de decisión. En la Sección 3., se aborda el desarrollo de la HiperHeurística propuesta. Se plantean los Principios de la Inteligencia de Grupo en los que se sustenta, las X formas de la MetaHeurística PSO MultiObjetivo, consideradas candidatas para componer su dominio, y la caracterización del Espacio de Búsqueda para la Clase de Problemas a Resolver. Adicionalmente, se establece la habilidad que cada forma X-FPSO exhibe para satisfacer, en mayor o menor grado, los cinco Principios de la Inteligencia de Grupo. Finalmente, con esta información, se propone el diseño de una Red Neuronal Artificial de Retropropagación, para que capte en sus patrones de entrada/salida, los aspectos identificados como característicos y que relacionan: Soluciones, formas X-FPSO y habilidad en la Inteligencia de Grupo y Espacio de Búsqueda en la Clase de Problemas abordados. Esta Red Neuronal aproximará una función que tendrá como patrón de salida o respuesta a cierto patrón de entrada, la forma X-FPSO más apta para continuar la estrategia de búsqueda, a partir de cierta instancia de decisión. En la Sección 4., son presentadas las conclusiones más relevantes y en la Sección 5., la extensa bibliografía consultada.

2. HEURÍSTICAS, METAHEURÍSTICAS E HIPERHEURÍSTICAS

2.1 Métodos Heurísticos

La etimología del término Heurística, proviene de la palabra griega *heuriskein* que se traduce como “encontrar”. En el contexto del presente trabajo, es necesaria una definición, en tanto Método Heurístico. Más allá de las múltiples interpretaciones y de la vastísima bibliografía al respecto, se propone aquí la idea de Método Heurístico sugerida en (DÍAZ *et al.*1996):

Un Método Heurístico constituye un procedimiento para resolver un problema de optimización bien definido, mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución.

Desde la definición propuesta, el método está dirigido a un problema específico, del cual se conocen sus características (bien definido), al efecto de sacar provecho de ellas en la estrategia de búsqueda concebida.

La pregunta que cabe hacerse es: ¿cómo poder aprovechar las similitudes que ciertos problemas, fundamentalmente de carácter combinatorio, exhiben, cuando un método Heurístico ha generado muy buenos resultados? Esta es la idea que subyace bajo el concepto de MetaHeurística.

2.2 Métodos MetaHeurísticos

El término MetaHeurística, deriva del complemento entre la palabra *heuriskein* y el prefijo *meta*, el cual se traduce como “más allá de” o “en un nivel superior de/a”. Fue introducido por primera vez por Fred Glover, al presentar el método (*MetaHeurístico*) denominado *Tabu Search* (Búsqueda Tabú) (GLOVER, 1989). Sin embargo, existe una controversia respecto de si una MetaHeurística es diferente de una Heurística, y varias fuentes pueden ser citadas en tal sentido. Según (BLACK, 2009): **a)** Una MetaHeurística define un Marco de Referencia Algorítmico cuyo Enfoque puede ser especializado para Resolver Problemas de Optimización; **b)** Una MetaHeurística constituye una Estrategia de Alto Nivel que Guía/Conduce Heurísticas en la Búsqueda de Soluciones Factibles. Si se toma en consideración **a)**, se tiene un Método Heurístico, entendido como un marco de referencia algorítmico. En cambio, según **b)**, se observa una respuesta a la pregunta referida. Siguiendo esta línea, se adoptará la siguiente definición (OSMAN Y KELLY, 1996):

Una MetaHeurística se define como un proceso iterativo que guía una Heurística subordinada, combinando diferentes conceptos para explorar y explotar las características que pueda exhibir el espacio de búsqueda.

Frecuentemente, al considerar la referencia al término proceso iterativo, se suele hablar de Algoritmo MetaHeurístico. En el presente trabajo, se utilizarán de manera indistinta, los términos MetaHeurística, Algoritmo MetaHeurístico o Método MetaHeurístico. Idénticamente para las Heurísticas.

2.3 Concepto de HiperHeurística

2.3.1 Introducción

Las MetaHeurísticas constituyen, en general, métodos especializados de búsqueda sobre un espacio de posibles soluciones de un problema, si bien su grado de especialización, es menos restringido que el de las Heurísticas, conforme las definiciones adoptadas desde el estado del arte. Resultan eficientes, en términos de su capacidad de localizar soluciones de buen nivel de aptitud, aún las extendidas al dominio de optimización multiobjetivo, dependiendo, en este último caso, de la MetaHeurística utilizada y de la función de aptitud para ponderar el mérito simultáneo en la satisfacción de los múltiples objetivos pretendidos.

Sin embargo, exhiben dos limitaciones importantes: **a)** Se requiere de un ajuste previo de parámetros según el tipo de problema a resolver, de manera que la imposibilidad de automatizar dicho ajuste, limita la extensión de la aplicación de una MetaHeurística, para la búsqueda de soluciones sobre Clases de Problemas. Suponen, en su mejor performance, un grado detallado relativo al conocimiento sobre el dominio del problema a resolver, para poder generar resultados de calidad (DOWSLAND, 1998), (AICKELIN Y DOWSLAND, 2000), (BURKE *et al.* 2001).

Tal situación redundante en la limitación referida, sobre la reutilización en otros dominios de aplicación e, inclusive, en instancias diferentes (un contexto de datos muy distinto, por caso) del mismo tipo de problema para la cual se especializan, mediante el ajuste externo de sus parámetros (GLOVER Y KOCHENBERGER, 2003); **b)** Por lo general el tiempo de cómputo requerido, resulta muy elevado, situación crítica si se pretendiese aplicar a problemas de optimización en tiempo real. Por tales motivos, las MetaHeurísticas son consideradas con frecuencia como técnicas falibles. Si bien, en trabajos previos, se ha podido reducir el impacto de la limitación indicada en **a)**, mediante la combinación de los principios de la Inteligencia de Grupo y los Algoritmos Evolutivos, y extendiendo sus capacidades de autoajuste de parámetros a las optimizaciones multiobjetivo a partir, por caso, de la MetaHeurística FEPSO GIST (SCHWEICKARDT *et al.* 2010), sobre el punto **b)** no se han producido importantes avances. De esta manera, la corriente de pensamiento en *Soft Computing*, relativa al campo de la optimización, comienza a plantearse soluciones que no revisten el carácter de óptimas. Aún en problemas determinísticos pues, si el contexto de decisión está dominado por incertidumbres, ha sido introducido y fundamentado el concepto de solución satisfactoria, como sustitución lógica de la solución óptima (SCHWEICKARDT y MIRANDA, 2010). Surge, de tal modo, el concepto de solución “suficientemente buena”, la cual permite proponer métodos de mayor generalidad para resolver diferentes tipos de problemas y, a su vez, que las soluciones puedan ser obtenidas requiriendo menores tiempos de cómputo. Este es el concepto que subyace en la idea primigenia de las Hiperheurísticas: Dado que diferentes Heurísticas simples, exhiben diferentes fortalezas y debilidades, una estrategia más eficiente en términos de la relación velocidad de respuesta/calidad de soluciones, la constituye su utilización conjunta, para que la ejecución oportuna de cada una pueda compensar las debilidades de las otras. La HiperHeurística, supondría una función, de diferente complejidad, según el caso, tal que examinando las propiedades de las soluciones obtenidas en el proceso de búsqueda y/o las características del espacio en el cual las mismas van presentándose, conmute, si procede, a una Heurística de mejor aptitud, como nuevo proceso a ejecutar en los siguientes pasos de evolución sobre la estrategia definida. Expresada esta idea programáticamente en pseudo-código, se tendría lo siguiente:

Si (CondiciónDelProblema(P))= C1) Entonces
 Aplicar Heurística1
 sino
 Si (CondiciónDelProblema(P))= C2) Entonces
 Aplicar Heurística2
 Sino...
 FinSi (en $C = \{C1, C2, \dots CH\}$)
 para el conjunto seleccionado de Heurísticas, $C = \{C1, C2, \dots CH\}$,
 sobre el problema a resolver, P.

2.3.2 Definición de HiperHeurística

Una HiperHeurística puede ser definida, entonces, como una estrategia de búsqueda tal, que, a partir de cierto conjunto pre-establecido de Heurísticas/MetaHeurísticas de bajo nivel, reconocidas como eficientes en la resolución de problemas pertenecientes a una misma clase, selecciona la que resulte más apta en cada instancia de decisión, identificada mediante algún proceso de aprendizaje, ejecutándola para el hallazgo de buenas soluciones en el menor tiempo posible. Tal estrategia continúa hasta que cierta condición de salida o finalización resulte satisfecha. Por tanto, una HiperHeurística constituye un Algoritmo Heurístico de alto nivel, identificado por el proceso de aprendizaje/selección referido. Sin embargo, no se la refiere como MetaHeurística, conforme la definición adoptada y presentada más arriba, puesto que existe una diferencia muy importante: la MetaHeurística, al igual que las Heurísticas, es aplicada en el espacio de soluciones del problema, mientras que la HiperHeurística es aplicada en cierto espacio de Heurísticas/MetaHeurísticas de bajo nivel. Es decir: una Hiperheurística trata con métodos de solución, mientras que una MetaHeurística trata con las soluciones de cierto problema. El nivel de abstracción de una HiperHeurística resulta, consecuentemente, mayor que el correspondiente a una MetaHeurística.

Es interesante una breve presentación sobre los orígenes y desarrollos del concepto de HiperHeurística: la primera publicación conocida sobre el tema data de 1961 (FISHER Y THOMPSON, 1961); luego, en las siguientes tres décadas (de los 60's a los 90's) se pueden encontrar muy pocos trabajos relacionados. El trabajo de Kitano (KITANO, 1990) parecería componer un punto de inflexión, orientado hacia el estudio actual sobre HiperHeurísticas, ya que en la pasada década se produjo un incremento significativo en el interés por este campo de investigación, debido, principalmente: a la posibilidad de tratar problemas reales complejos desde la investigación operativa, a los avances en *Soft Computing*, y en el *hardware* soporte de las implementaciones de los mismos. Paralelamente, han proliferado por la misma razón, métodos de búsqueda Heurística (OSMAN Y LAPORTE, 1996) y MetaHeurística, pudiendo referirse una gran cantidad de técnicas, entre las que se encuentra la PSO y todas sus variantes tratadas por los autores del presente escrito, entre otras.

Las MetaHeurísticas evolucionaron procurando minimizar la necesidad de ajustar externamente sus parámetros, por caso, la EPSO o su extensión al dominio multiobjetivo FEPSO (SCHWEICKARDT Y MIRANDA, 2010). Paralelamente, lo hicieron las HiperHeurísticas, avanzando en la rapidez y generalización sobre las soluciones posibles de obtener en el dominio de Clases de Problemas. Ambas estrategias, Meta e HiperHeurísticas, parecerían perseguir, actualmente, la misma finalidad. De modo que pueden observarse en el estado del arte, algunas propuestas, como la que en este trabajo es presentada, y que combinan, en el contexto de la definición de HiperHeurística, lo mejor de ambas estrategias. Por supuesto, constituye un campo vastísimo de investigación, con enorme potencial.

2.3.3 Métodos de Aprendizaje/Selección en las HiperHeurísticas

Uno de los puntos básicos de una estrategia HiperHeurística estriba en el desarrollo de métodos de aprendizaje/selección, capaces de identificar cuáles de las Heurísticas/MetaHeurísticas de bajo nivel pre-establecidas resultan más aptas, conforme las instancias de decisión inherentes a la búsqueda de soluciones para el problema abordado. Desde la bibliografía consultada, y atendiendo a tales métodos, se propone aquí la siguiente clasificación de HiperHeurísticas:

A) HiperHeurísticas Sin Aprendizaje: las cuales pueden dividirse en A-1) Aleatorias y A-2) *Greedy (Codiciosa)*. Se presentan en (COWLING *et al.* 2000), (COWLING *et al.* 2001) y (COWLING *et al.* 2002-a); B) HiperHeurísticas Con Aprendizaje: las cuales admiten la subclasificación siguiente: B-1) Mediante Función de Elección y B-2) Mediante MetaHeurísticas; C) HiperHeurísticas Basadas en Razonamiento y D) HiperHeurísticas Con Aprendizaje Incremental.

Se subclasifican, si procede, y describen, contemplando *la estrategia*, el *tipo de método* y el *criterio de mejora en las soluciones*, a continuación:

A-1) HiperHeurísticas Sin Aprendizaje Aleatorias: Se observan tres *tipos de estrategias*:

A-1-1) Aleatoria Simple: La estrategia consiste en elegir la Heurística de bajo nivel en forma aleatoria, desde una distribución de probabilidad uniforme, y el método de aplicación puede acotarse sólo para aquellas Heurísticas que generen movimientos en el espacio de búsqueda, tales que aporten mejoras en las soluciones obtenidas (criterio: sólo mejora), o aceptar cualquier movimiento y, por tanto, cualquier Heurística, durante un ciclo de iteraciones pre-establecido (criterio: aceptación de movimiento); A-1-2) Aleatoria Descendente: La estrategia consiste en seleccionar, también aleatoriamente y desde una distribución uniforme de probabilidades, una Heurística de bajo nivel, y el método supone su aplicación hasta que la misma no aporte mejoras en las soluciones obtenidas; A-1-3) Permutación Aleatoria: La estrategia consiste en

generar aleatoriamente (se omite ya mencionar el empleo de una distribución uniforme de probabilidades) una permutación del vector de Heurísticas, y ejecutarlas en el orden obtenido de la misma forma que en la estrategia Aleatoria Simple; A-1-4) Permutación Aleatoria Descendente: la estrategia es igual a la anterior, y el método y criterio para la ejecución de las Heurísticas se corresponde con el de la Aleatoria Descendente; A-2-1) HiperHeurísticas Sin Aprendizaje Greedy: esta estrategia, supone la selección de cada Heurística, mediante un método que evalúa su aptitud, y se la emplea bajo el criterio sólo mejora, es decir, hasta que no produce mejoras en las soluciones alcanzadas, conmutando a otra Heurística del conjunto pre-establecido.

B-1) HiperHeurísticas Con Aprendizaje Mediante Función de Elección: se identifican los siguientes tipos: B-1-1) Agrupadas (COWLING *et al.* 2000), (COWLING *et al.* 2001), (COWLING *et al.* 2002-a) y (COWLING *et al.* 2002-b): La estrategia consiste en seleccionar la Heurística de aplicación, como el nombre lo indica, mediante una función de elección que analiza, por pares, el rendimiento de cierta Heurística candidata H_k , comparándola con otra del conjunto, H_j , $j \neq k$, mediante la siguiente optimización: si n_h es el número de Heurísticas del conjunto pre-establecido, entonces:

$$\begin{aligned} & \text{Maximizar } \forall j \text{ en } [1..n_h], j \neq k \\ & \alpha \times f_1(H_k) + \beta \times f_2(H_k, H_j) - \gamma \times f_3(H_k) \end{aligned} \quad (1)$$

siendo f_1 una función que pondera la mejora reciente de la Heurística H_k , f_2 una función que pondera la mejora relativa de las dos Heurísticas (H_k , H_j) aplicadas secuencialmente, y f_3 es una función que representa el tiempo en el cual la Heurística H_k no ha sido aplicada, a través del número de iteraciones desde la última vez en que se empleó; α , β y γ son parámetros definidos externamente. Dependiendo de las características funcionales, el problema puede plantearse como una minimización.

Se tienen dos versiones: *con y sin ajuste* de tales parámetros, a medida que evoluciona la estrategia de búsqueda; B-1-2) Múltiples Objetivos (COWLING *et al.* 2001), (COWLING *et al.* 2002-a), (COWLING *et al.* 2002-b) y (BURKE *et al.* 2003-a): en esta estrategia, la función de elección supone definido un conjunto de L subobjetivos a satisfacer y en ellos se establecen, en general, diferentes características funcionales para f_1 , f_2 y f_3 en (1), así como parámetros α , β y γ . De manera que la selección de la Heurística candidata H_k se realiza, en las mismas condiciones que la anterior, (1), mediante la siguiente optimización:

$$\begin{aligned} & \text{Maximizar/Minimizar } \forall j \text{ en } [1..n_h], j \neq k, \forall l \text{ en } [1..L] \\ & \sum_{l=1}^L \{ \alpha_l \times f_{1l}(H_k) + \beta_l \times f_{2l}(H_k, H_j) - (\gamma_l/L) \times f_{3l}(H_k) \} \end{aligned} \quad (2)$$

siendo l el l -ésimo subobjetivo en el conjunto L

B-2) HiperHeurísticas Con Aprendizaje Mediante MetaHeurísticas: en este caso, la estrategia se sustenta en la aplicación de una MetaHeurística de alto nivel, capaz de evolucionar en el espacio de Heurísticas (o MetaHeurísticas) de bajo nivel, mejorando, en cierta instancia de decisión, el método abordado para la búsqueda en el espacio de soluciones. Las propuestas son diversas, así como el tipo de MetaHeurística empleada. Son citadas algunas: B-2-1) Tabu Search: o Búsqueda Tabú (BURKE *et al.* 2003-a), pero se preserva su nombre original, pues así es difundida en la bibliografía, aún la escrita en español; B-2-2) Ant Colony: o Colonia de Hormigas (BURKE *et al.* 2003-b) y (BURKE *et al.*, 2003-c); B-2-3) Simmulated Aneanning: o Recocido Simulado (BAI Y KENDALL, 2006); B-2-4) Computación Evolucionaria: utilizando diferentes esquemas híbridos de selección evolucionaria, mediante operadores evolutivos sobre la población de reglas Heurísticas (SMITH, 2002) y (BURKE *et al.* 2003-b). Este enfoque también ha recibido el nombre de soporte en algoritmos meméticos; y B-2-5) Algoritmos Genéticos: cuya estrategia se describirá brevemente, por encontrarse como la más difundida y representativa desde las HiperHeurísticas soportadas en MetaHeurísticas. Las HiperHeurísticas basadas en Algoritmos Genéticos tienen como propopósito evolucionar sobre el método de solución, y no sobre las soluciones directamente, tarea que le corresponde a las Heurísticas (MetaHeurísticas) de bajo nivel, como se dijo. Típicamente en el Algoritmo Genético, el cromosoma representa una lista de Heurísticas que pueden ser aplicadas sobre la solución actual. Resulta de interés citar el trabajo de Kitano (KITANO, 1990) quien fue uno de los primeros en abordar este tipo de HiperHeurísticas, desarrollando un Algoritmo Genético, para el diseño de una Red Neuronal Artificial (se omitirá, dando de aquí en más por sobreentendido, que es Artificial). En el diseño de Redes Neuronales basado en tales algoritmos, generalmente el cromosoma representa las configuraciones posibles de las redes (número de capas ocultas y de neuronas por capa, esencialmente) mediante una codificación directa en el espacio de Redes Neuronales. Kitano, en cambio, propuso una codificación indirecta, que denominó método de codificación gramática, el cual codifica una serie de reglas o Heurísticas que generan las Redes Neuronales. De modo que el Algoritmo Genético se aplicaba en el dominio de tales Heurísticas, y las Heurísticas seleccionadas por la función de aptitud, solidaria al Algoritmo Genético, se aplicaba en el dominio de las configuraciones de Redes Neuronales. De tal Manera, por definición, resulta una HiperHeurística soportada en Algoritmos Genéticos. Los resultados obtenidos por el trabajo de Kitano fueron superiores a los obtenidos por los métodos de codificación directa, y esta estrategia HiperHeurística, fue extendida a diversas clases de problemas. Otras estrategias que presentan buenos resultados en la forma de HiperHeurísticas basadas en Algoritmos Genéticos pueden consultarse en (FANG Y ROSS, 1994), (DORNDORF Y PESCH, 1995), (TCHEPRASOV Y GOODMAN, 1996) y (NORENKOW Y GOODMAN, 1997).

C) HiperHeurísticas Basadas en Razonamiento: (PETROVIC Y QU, 2002), (BURKE *et al.* 2002), la estrategia que se aplica en este tipo de HiperHeurística, consiste en emplear un conjunto de instancias de decisión/vector de soluciones del problema abordado, para el entrenamiento, por caso, de una Red Neuronal, a los efectos de la detección de patrones de estado/solución u otros elementos pertinentes, que permitan evaluar la aptitud de la Heurística que se está aplicando. Posteriormente, los patrones salida, son considerados en forma exclusiva o complementada con nuevos patrones de entrada, como las instancias anteriores de decisión/vector de soluciones, sobre nuevas instancias de decisión y, con ello, se logra evolucionar sobre el espacio de Heurísticas. Comúnmente, el criterio de selección adoptado para las soluciones obtenidas, es el referido en las HiperHeurísticas Aleatorias Descendentes, como sólo mejora. No obstante, es claro que puede ser propuesto otro criterio, dada la apertura que esta estrategia exhibe, tal y como se ha realizado, en parte, dentro de los aportes que intenta ofrecer el presente trabajo.

D) HiperHeurísticas Con Aprendizaje Incremental: la estrategia consiste en, a partir de *instancias de entrenamiento*, obtenidas en forma similar a la descrita en la HiperHeurística precedente, deducir iterativamente un conjunto de reglas, las cuales, en instancias de decisión posteriores, pueden ser aplicadas para resolver nuevas instancias. La diferencia respecto de la HiperHeurística Basada en Razonamiento, estriba en la posibilidad de generar nuevas reglas Heurísticas, acotadas por ciertos criterios, que permitan satisfacer las cualidades establecidas en la definición de una HiperHeurística. Específicamente, la rapidez en la obtención de buenas soluciones. Un ejemplo de tal estrategia, la constituye la HiperHeurística XCR propuesta en (ROSS *et al.* 2002).

3. LA HIPERHEURÍSTICA PROPUESTA: HY X-FPSO CBR

3.1 Objetivo del Diseño de la Estrategia HY X-FPSO CBR

El diseño de una HiperHeurística cuya estrategia esté identificada por un dominio de MetaHeurísticas, variantes de la FPSO, X-FPSO, capaces de resolver problemas de optimización MultiObjetivo, responde a la experiencia acopiada por parte de los autores en el empleo y desarrollo de propuestas para tales variantes. Adicionalmente, el tipo de problemas que deben resolverse en la aplicación, para la cual esta HiperHeurística se diseña, constituye una Clase de Problemas, que se puede caracterizar, fundamentalmente, por el Espacio de Búsqueda en sus soluciones.

Las cualidades de cada solución o partícula del enjambre, permiten identificar tres aspectos, lo cuales constituyen el soporte para la información requerida, en la definición de un mecanismo de selección entre las formas X-

FPSO pre-definidas, para cierta instancia de decisión del Algoritmo HiperHeurístico, de la variante X más apta. Los tres aspectos son: **a)** La relación entre los cinco Principios de la Inteligencia de Grupo que debe satisfacer cualquier MetaHeurística que opere con una población de partículas, **b)** La relación entre el Espacio de Búsqueda caracterizado por la Clase de Problemas a resolver y el Vector Solución obtenido para cierta instancia de decisión y **c)** La habilidad diferenciada que cada forma X-FPSO, exhibe para satisfacer algunos de estos cinco Principios, mejor que el resto.

Los tres aspectos, como pudo observarse en múltiples simulaciones, trabajan en forma complementaria. De manera que el objetivo de un Algoritmo HiperHeurístico tal como el HY X-FPSO CBR propuesto, consiste en brindar soluciones de muy buena calidad, en forma rápida, comparada con la aplicación de una única forma o variante X-FPSO, mediante un diseño de su estrategia capaz de captar la información sobre estos tres aspectos, en las instancias de decisión más adecuadas. A continuación se presentan los conceptos y desarrollos necesarios en tal diseño.

3.2 Principios de la Inteligencia de Grupo (IG) (*Swarm Intelligence*)

La Inteligencia de Grupo, IG, (o *Swarm Intelligence*) refiere un *tipo* de inteligencia artificial, basado en el comportamiento de sistemas colectivos, auto-organizados. La expresión fue introducida en 1989 por Gerardo Beni (BENI Y WANG, 1989). El término *swarm* describió, a su entender, el comportamiento de los autómatas celulares, que exhiben características similares, a las observables en ciertos sistemas biológicos, tales como los *insectos*. De esas características, mencionó: la descentralización, no-sincronización y simplicidad en los movimientos de los miembros del grupo. Además este término tiene una significación que lo trasciende: permite analizar el comportamiento social de grupos de individuos (peces, insectos, pájaros, etc.), identificando el tipo de configuración en sus sistemas de cooperación y auto-adaptación, al efecto de encontrar la manera óptima de alcanzar un objetivo. La comunicación, según los biólogos (KENNEDY Y EBERHART, 1995) constituye el parámetro clave en este proceso, de modo que a las cualidades mencionadas debe incorporársele la cooperación entre los individuos del *swarm*. Desde tales consideraciones, pueden ser establecidos cinco principios que caracterizan la IG (*Swarm Intelligence Principles*). Estos principios, y su breve enunciación, resultan: **1) Proximidad:** Promueve la habilidad que debe exhibir el grupo, para ejecutar cálculos simples de espacio y tiempo en sus movimientos hacia el objetivo pretendido; **2) Calidad:** Promueve la habilidad del grupo para responder a los factores que induzcan mejoras en la aptitud de sus individuos, en el espacio de soluciones; **3) Diversidad de Respuesta:** Promueve la posibilidad de que los individuos tengan respuestas diferentes ante los mismos estímulos; **4) Estabilidad:** Promueve la habilidad de que el grupo permanezca estable, en ausencia de estímulos que induzcan mejoras en las soluciones alcanzadas; y **5) Adaptación:** Constituye un aspecto complementario de la **Estabilidad**, puesto

que promueve la habilidad de que los individuos reaccionen ante cualquier cambio en la aptitud de las soluciones alcanzadas.

3.3 La MetaHeurística PSO y sus Formas o Variantes X-PSO MultiObjetivo o X-FPSO

3.3.1 Requerimientos para la Extensión Multiobjetivo de la MetaHeurística PSO: FPSO

El enfoque adoptado en los desarrollos de las MetaHeurísticas más difundidas en la bibliografía especializada fue primigeniamente MonoObjetivo. Existen diferentes propuestas en el estado del arte, para extender ciertas MetaHeurísticas al dominio MultiObjetivo.

Específicamente, para la PSO, se tienen propuestas como la presentada en (COELLO Y SALAZAR, 2002), denominada MOPSO (MultiObjective PSO). Al efecto de concebir una función que pondere la aptitud de las soluciones en un contexto MultiObjetivo de optimización, deben satisfacerse *dos requerimientos* (SCHWEICKARDT Y MIRANDA, 2010): **A) Pareto-Dominancia en las soluciones:** las soluciones igualmente óptimas deben satisfacer una escala pareto-dominante, lo cual supone que si dos soluciones componen un frente pareto-óptimo, ninguna de ellas puede ser mejorada en cierto objetivo, sin empeorar, al menos, en uno de los restantes (soluciones no dominadas), y resultan igualmente preferibles o aptas. Por el contrario, dos soluciones exhibirán dominancia en este sentido, si una de las dos mejora, respecto de la otra, cierto objetivo, sin empeorar el resto. La dominante resultará, así, preferible o más apta; **B) Compatibilidad Métrica respecto del Espacio de Búsqueda:** en general, aún para los Algoritmos MetaHeurísticos MonoObjetivo, el diseño de una función de aptitud que no introduzca una métrica adecuada para definir el mérito de las soluciones obtenidas, evidenciará ciertas limitaciones que, erróneamente, pueden asociarse con el algoritmo o con la especificación de los parámetros en el modelo formal. Este efecto, se pronuncia aún más al extender la MetaHeurística considerada al contexto MultiObjetivo. El Espacio de Búsqueda, tiene una influencia sustancial. Múltiples variables de decisión, discretas, con intervalos estrechos en sus valores factibles, y buenas soluciones muy dispersas en tal espacio, impactan fuertemente sobre la compatibilidad que debe guardar la métrica de la función de aptitud, respecto de los estímulos que induce sobre los agentes, para explorar eficientemente el espacio de búsqueda. Este aspecto, ha sido muy estudiado para los aportes del presente trabajo. A modo de síntesis, puede decirse que: para las MetaHeurísticas sustentadas en la IG, una función de aptitud resultará métricamente compatible con el espacio de búsqueda, en la medida que pueda propiciar el cumplimiento de los cinco principios enunciados.

Otro aspecto que debe ser mencionado, particularmente en la Clase de Problemas que debe resolver la HiperHeurística propuesta, analizado por los autores como objetivos individuales aplicando variantes de la MetaHeurística

FPSO, son las incertidumbres no estocásticas de valor (SCHWEICKARDT Y MIRANDA, 2010), que siguen el paradigma de riesgo planteado por Keynes (LAVOIE, 1992). Las mismas se presentan en las magnitudes de cada criterio cuyo mérito quiere maximizarse en la optimización. Los dos requerimientos anteriores adscriben, implícitamente, a un paradigma en el cual se tiene certeza respecto las preferencias entre los objetivos, o bien sobre el grado de satisfacción que el alcance de cierto objetivo individual, produce en el tomador de decisiones. Adicionalmente, en la teoría de preferencias, tal paradigma reconoce sólo incertidumbres de naturaleza estocástica. Los procesos de toma de decisión, conforme este paradigma, se establecen en un entorno de riesgo tal, que puede representarse por algún conjunto equivalente de situaciones de certeza.

En última instancia, esto implica sostener que, en el Universo de Decisión, todos los estados de la naturaleza y las posibles alternativas, son susceptibles de modelar mediante alguna distribución de probabilidades. De forma tal que a las dos condiciones precedentes, **Pareto-Dominancia** y **Compatibilidad Métrica**, se le agrega una tercera: **C) Capacidad de Captar las Incertidumbres de Valor**: posibilidad de modelar las incertidumbres en el grado de satisfacción asociado al cumplimiento de un objetivo individual, o a las preferencias entre objetivos. La alternativa propuesta por los autores para tratar con la tercer condición o requerimiento y que permite operacionalmente extender la forma PSO a su versión MultiObjetivo, FPSO, consiste en modelar cada variable asociada a los mismos mediante Conjuntos Difusos (SCHWEICKARDT Y MIRANDA, 2010). Como consecuencia, la función de aptitud de la MetaHeurística PSO, resultará también un Conjunto Difuso, resultado de algún operador pertinente entre los Conjuntos Difusos asociados a las variables correspondientes a cada Objetivo.

3.3.2 Formulación de la Función de Aptitud Difusa para la MetaHeurística FPSO

Para formalizar la extensión FPSO, se propone aplicar el principio de toma de decisión en ambientes difusos, formulado por Bellman y Zadeh (BELLMAN Y ZADEH, 1970). Primeramente se dirá que un Conjunto Difuso, constituye una función, denominada función de pertenencia, sobre cierta variable real (discreta o continua), que le asocia a cada valor en su dominio, otro valor llamado de aceptación, satisfacción o pertenencia, en el intervalo $[0, 1]$. El valor 1 corresponde a la máxima pertenencia. Entonces el principio de Bellman y Zadeh, puede sintetizarse como sigue:

Considérese un *conjunto de* objetivos difusos (esto es: sus incertidumbres de valor son representadas por medio de conjuntos difusos): $\{\mathbf{O}\} = \{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_N\}$, cuyas *funciones de pertenencia* resultan ser μ_{O_j} , con $j=1\dots N$, y un conjunto de restricciones difusas (esto es: sus incertidumbres de valor en los límites superior e inferior sobre las variables de decisión, son representadas

por conjuntos difusos): $\{R\} = \{R_1, R_2, \dots, R_H\}$, cuyas funciones de pertenencia resultan ser μ_{R_i} , con $i=1\dots H$.

Se denomina Conjunto Difuso de Decisión, al obtenido mediante:

$$D = O_1 \langle C \rangle O_2 \langle C \rangle \dots \langle C \rangle O_N \langle C \rangle R_1 \langle C \rangle R_2 \langle C \rangle \dots \langle C \rangle R_H \quad (3)$$

donde $\langle C \rangle$ es un *operador entre Conjuntos Difusos* que recibe el nombre de confluencia (por caso, la *intersección*). Asociado al operador $\langle C \rangle$ entre los *conjuntos difusos*, existe un *operador C* entre sus funciones de pertenencia, que genera, desde (3), el valor de pertenencia del Conjunto Difuso de Decisión, conforme los valores individuales de las funciones de pertenencia del segundo miembro. Es decir:

$$\mu_D = \mu_{O_1} C \mu_{O_2} C \dots C \mu_{O_N} C \mu_{R_1} C \mu_{R_2} C \dots C \mu_{R_H} \quad (4)$$

El *operador C*, recibe el nombre general de *t-norma*. Por ejemplo, si la confluencia fuese la intersección, $\langle C \rangle \equiv \cap$, *C* resulta la *t-norma min*: el mínimo valor, para cierta instancia de las variables de decisión, en el conjunto de funciones de pertenencia del segundo miembro de la expresión (4). Entonces, si $\{A\}$ es un conjunto de alternativas sobre las que debe decidirse por la mejor, en términos del *modelo objetivo-restricciones* $\{O\}$ - $\{R\}$, se define como decisión maximizante de Bellman y Zadeh, al valor de la función de pertenencia en el conjunto de decisión difusa, dado por:

$$\mu_D^{Max} = \text{MAX}^{[A]} \{ \mu_{O_1} C \mu_{O_2} C \dots C \mu_{O_N} C \mu_{R_1} C \mu_{R_2} C \dots C \mu_{R_H} \} \quad (5)$$

Nótese que todos los Conjuntos Difusos (objetivos y restricciones) son mapeados en el mismo Conjunto Difuso de Decisión **D**, tratándose de la misma forma. Respecto del concepto de *t-norma*, es definida por las siguientes propiedades: Si $t: [0, 1] \rightarrow [0, 1]$ es una *t-norma*, entonces: a) $t(0,0) = 0$; $t(x,1) = x$; b) $t(x,y) = t(y,x)$; c) if $x \leq \alpha$ e $y \leq \beta \Rightarrow t(x,y) \leq t(\alpha,\beta)$; and d) $t((t(x,y),z)) = t(x,t(y,z))$. De modo que la aplicación de tal principio para construir la función de aptitud MultiObjetivo, en un contexto de incertidumbre de valor, seguirá los siguientes pasos: **1)** Cada Objetivo y cada Restricción son representados por Conjuntos Difusos, que captan e introducen tales incertidumbres; **2)** La función de aptitud difusa, $fapD$, resultará de la decisión maximizante dada por la expresión (5). Con ello se satisface el requerimiento **C** (Captación de Incertidumbres de Valor) y **3)** Se debe definir una *t-norma* que satisfaga, con $\mu_D^{Max} \equiv fapD$, los requerimientos **A**) (Pareto Dominancia) y **B**) (Compatibilidad Métrica) referidos en 3.3.1.. Para el Espacio de Búsqueda que caracteriza a la Clase de Problemas MonoObjetivo, la *t-norma* que más se ajustó a los requerimientos de Pareto Dominancia y Compatibilidad Métrica, se denomina Producto de Einstein, y se define como:

$$t_{PE} = x \times y / (2 - (x + y - x \times y)) \quad (6)$$

Sobre cada objetivo/restricción, los Conjuntos Difusos solidarios a sus variables, se definen del siguiente modo: Considérense, dos límites, superior e inferior, en los valores posibles de la variable correspondiente a cierto objetivo/restricción m , v_m . Se referirán como $vMax_m$ y $vMin_m$, respectivamente. Adicionalmente, sea p_μ^m su *ponderador exponencial*, cuyo efecto sobre el conjunto difuso solidario, es su *contracción* ($p_\mu^m > 1$), aumentando la importancia del objetivo/restricción en la confluencia $\langle C \rangle$, o *dilatación* ($p_\mu^m < 1$), disminuyendo dicha importancia. Entonces, la función de pertenencia adoptada, de característica base lineal (este es el caso cuando $p_\mu^m = 1$), genérica para el objetivo/restricción m -ésimo, resultará de la expresión condicional: $\forall m$ en $[1..M]$, con M variables objetivo/restricción:

$$\mu_m = 1 \quad ; \text{ si } vMin_m \geq v_m \quad (7)$$

$$\mu_m = \left(\frac{(vMax_m - v_m)}{(vMax_m - vMin_m)} \right)^{p_\mu^m} \quad ; \text{ si } vMin_m \leq v_m \leq vMax_m \quad (8)$$

$$\mu_m = 0 \quad ; \text{ si } vMax_m \leq v_m \quad (9)$$

Luego, la función de aptitud difusa $fapD$, estará dada por la decisión estática maximizante en cada instancia de la estrategia, según la expresión (6):

$$fapD = \mu_D^{Max} = t_{PE}^{Max} \{ \mu_{m1}; \mu_{m2}; \dots; \mu_{mM} \} \quad (10)$$

3.3.3. Variantes o Formas X-FPSO

A) X1: Forma Canónica de la MetaHeurística PSO: Se presentan brevemente los desarrollos extraídos de (SCHAWWEICKARDT Y MIRANDA, 2010) correspondientes a la MetaHeurística PSO. Desde una iteración a la siguiente, cada partícula se mueve en el *espacio de búsqueda*, conforme a cierta *regla de movimiento* que depende de tres factores, que se explican a continuación. Se indicará mediante $[p]$ el vector de partículas en movimiento, de modo que p_i resultará una partícula individual de las n que pertenecen al enjambre. Adicionalmente, se indicará mediante $[b]$ el vector de las mejores posiciones (cada posición es, a su vez, un vector) que las partículas han alcanzado individualmente en las iteraciones anteriores (aspecto referido en el modelo como *vida pasada* de la partícula); entonces b_i se corresponderá con el óptimo individual de la partícula i en su vida pasada. Del mismo modo, se indicará mediante $[b]_G$ el vector cuyos elementos son las mejores posiciones globalmente *alcanzadas* por el conjunto de partículas en las iteraciones anteriores; entonces se indicará mediante b_G al óptimo global alcanzado por el sistema de partículas hasta la iteración presente. Dado el vector que indica la posición de las partículas en cierta iteración k , X^k , el cambio de posición en la iteración siguiente, $k+1$, para la partícula i -ésima, resultará de la siguiente regla de movimiento:

$$\mathbf{X}^{[k+1]}_i = \mathbf{X}^{[k]}_i + \mathbf{V}^{[k+1]}_i \times \Delta t \quad (11)$$

donde el término $\mathbf{V}^{[k+1]}_i$ es referido como velocidad de la partícula i -ésima; Δt es el paso de iteración (simil *temporal*) e igual a la unidad. De modo que es más frecuente encontrar la expresión:

$$\mathbf{X}^{[k+1]}_i = \mathbf{X}^{[k]}_i + \mathbf{V}^{[k+1]}_i \quad (12)$$

El vector velocidad para la partícula i -ésima, se expresa como sigue:

$$\mathbf{V}^{[k+1]}_i = \mathbf{V}^{[k]}_i + w_C \times (r_1^{[k]}) \times [\mathbf{b}^{[k]}_i - \mathbf{X}^{[k]}_i] + w_S \times (r_2^{[k]}) \times [\mathbf{b}^{[k]}_G - \mathbf{X}^{[k]}_i] \quad (13)$$

El primer término de la suma (13), representa la inercia o hábito de la *partícula* i : tiende a mantener su movimiento, para la iteración $k+1$, en la dirección en la que se movía en la iteración k . El segundo término representa la memoria o capacidad cognitiva de la partícula i : es atraída por el mejor punto del espacio de búsqueda alcanzado individualmente en su vida pasada; y el tercer término representa la cooperación entre el conjunto, o capacidad social, de la partícula i respecto del enjambre: *las* partículas comparten información sobre la mejor posición globalmente alcanzada por el enjambre. La incidencia de estos *factores* sobre cada partícula, está dada por las constantes o parámetros del modelo, $w_{C,S}$. El parámetro w_C recibe el nombre de constante cognitiva y el parámetro w_S se denomina constante social del enjambre. Intervienen los parámetros (r_1) y (r_2) , los cuales son números aleatorios uniformemente distribuidos en $[0,1]$, $U[0,1]$, y cuyo objetivo es emular el comportamiento estocástico (un tanto impredecible), que exhibe la población o enjambre, en cada iteración k .

La extensión FPSO MultiObjetivo, se plantea tal y como ha sido descrita en el epígrafe anterior, y la calidad de la solución en cierta instancia/iteración k , se pondera mediante el valor de la función de aptitud difusa, $fap\mathbf{D}_{[k]}$, computada según (10), sobre el vector posición $\mathbf{X}_{[k]}$. Esta descripción se adoptará como implícita para las variantes X-PSO MultiObjetivo \equiv X-FPSO restantes.

B) X2: PSO con Función de Decaimiento Inercial: En esta forma, el operador velocidad, dado por la expresión (10), es modificado mediante la introducción de una función decreciente con el número de iteraciones, k , denominada Función de Inercia o de Decaimiento Inercial, $\delta(k)$. Su objeto es reducir, progresivamente, la importancia del término de inercia. La expresión (10) es modificada sustituyendo el primer término por el producto entre una nueva constante, w_i , denominada constante de inercia y tal función $\delta(k)$:

$$\mathbf{V}^{[k+1]}_i = \delta(k) \times w_i \times \mathbf{V}^{[k]}_i + w_C \times (r_1^{[k]}) \times [\mathbf{b}^{[k]}_i - \mathbf{X}^{[k]}_i] + w_S \times (r_2^{[k]}) \times [\mathbf{b}^{[k]}_G - \mathbf{X}^{[k]}_i] \quad (14)$$

Una forma típica para $\delta(k)$, está dada por la siguiente expresión lineal:

$$\delta(k) = w_{Max} - k \times (w_{Max} - w_{Min}) / nTK \quad (15)$$

donde k es la *iteración actual*; nTK es el número máximo de iteraciones de la MetaHeurística; y $[w_{Min}, w_{Max}]$ son dos constantes inerciales, máxima y mínima, cuyos valores típicos resultan ser 0,4 y 0,9, respectivamente.

C) X3: PSO con Factor de Constricción: En este caso, el operador velocidad canónico se modifica como sigue:

$$\mathbf{V}^{[k+1]}_i = \chi \times \{ \mathbf{V}^{[k]}_i + \varphi_M \times (r_1^{[k]}) \times [\mathbf{b}^{[k]}_i - \mathbf{X}^{[k]}_i] + \varphi_C \times (r_2^{[k]}) \times [\mathbf{b}^{[k]}_G - \mathbf{X}^{[k]}_i] \} \quad (16)$$

donde χ se denomina Factor de Constricción, y se obtiene desde la siguiente expresión condicionada:

$$\chi = 2 \times \kappa / \left| 2 - \varphi - \sqrt{\varphi^2 - 4 \times \varphi} \right| \quad (17)$$

$$\text{con: } \varphi_M + \varphi_C = \varphi; \varphi > 4 \text{ y } 0 < \kappa \leq 1 \quad (18)$$

D) X4: EPSO (Evolutionary Particle Swarm Optimization): Combina la programación evolucionaria mediante operadores evolutivos similares a los establecidos para los Algoritmos Genéticos (particularmente: mutación, selección, combinación) con el movimiento PSO, aplicándolos sobre sus parámetros y óptimo global en la instancia de iteración k -ésima. La ecuación evolutiva para la regla del movimiento en el EPSO, se sintetiza en la velocidad de la partícula i -ésima:

$$\mathbf{V}^{[k+1]}_i = w_{il}^{*[k]} \times (r_1^{[k]}) \times \mathbf{V}^{[k]}_i + w_{iC}^{*[k]} \times (r_2^{[k]}) \times [\mathbf{b}^{[k]}_i - \mathbf{X}^{[k]}_i] + w_{iS}^{*[k]} \times (r_3^{[k]}) \times [\mathbf{b}^{[k]*}_G - \mathbf{X}^{[k]}_i] \quad (19)$$

donde: el superíndice (*), significa que los parámetros son evolutivos, producto de la mutación. La regla de mutación aplicable a las constantes $w_{[i], c, s]}$ tiene, como expresión general:

$$w_{[i], c, s]}^{*[k+1]} = w_{[i], c, s]}^{[k]} \times [1 + \sigma \times N(0,1)] \quad (20)$$

en la cual: σ es un parámetro de aprendizaje, externamente fijado, que controla la amplitud de las mutaciones; $N(0,1)$ es una variable aleatoria con distribución gaussiana de media 0 y varianza 1. Adicionalmente, el óptimo global también es perturbado en cada iteración, según la regla:

$$\mathbf{b}^{[k+1]*}_G = \mathbf{b}^{[k]*}_G + w_{iN}^{*[k]} \times N(0,1) \quad (21)$$

donde $w_{iN}^{*[k]}$, es un parámetro estratégico de la regla de movimiento, que controla la amplitud del vecindario de $\mathbf{b}^{[k]*}_G$ donde es más probable localizar la mejor solución global (entendida, al menos, como una solución mejor que la

$\mathbf{b}_G^{[k]*}$ actual, para la iteración k). El nuevo parámetro, como se indica con el superíndice (*), es también mutado según la regla (20).

E) X5: EPSO GIST (Global Individual Star/Topology): esta variante evolucionaria con Topología Estrella Global/Individual para la comunicación entre las partículas del enjambre, es presentada en (SCHWEICKARDT *et al.* 2010). La MetaHeurística EPSO GIST, tiene por ecuación evolutiva de movimiento, la expresión:

$$\mathbf{V}_i^{[k+1]} = w_{i1}^{*[k]} \times (r_1^{[k]}) \times \mathbf{V}_i^{[k]} + w_{iC}^{*[k]} \times (r_2^{[k]}) \times [\mathbf{b}_i^{[k]} - \mathbf{X}_i^{[k]}] + w_{iS}^{*[k]} \times (r_3^{[k]}) \times [\mathbf{b}_G^{[k]*} - \mathbf{X}_i^{[k]}] \times \alpha OG_i^{[k]} + \sum_{j=1, j \neq i}^n (\delta(k) \times w_{iC}^{*[k]} \times [\mathbf{b}_j^{[k]} - \mathbf{X}_i^{[k]}] \times \alpha OI_{ij}^{[k]}) \quad (22)$$

donde: $\alpha OG_i^{[k]}$: es la probabilidad de que el Óptimo Global (OG) afecte al movimiento de la partícula i, en la iteración k y $\alpha OI_{ij}^{[k]}$ es la probabilidad de que el Óptimo Individual (OI) de la partícula j, afecte al movimiento de la partícula i, $j \neq i$; $\delta(k)$ es el mismo factor multiplicativo empleado en (14), el cual va haciendo “decaer” la influencia de los OI, con probabilidades distintas de 0, conforme aumenta el número de iteraciones.

3.4 Satisfacción de los Principios de la IG para las Variantes X-FPSO y Caracterización del Espacio de Búsqueda

Como se dijo, a los efectos de poder diseñar la estrategia para que la HiperHeurística aquí propuesta, explore las mejores X-FPSO formas en cierta instancia de decisión, se captará el grado que cada X forma exhibe en la satisfacción de uno o más Principios de IG. Este es un hecho observado por los autores, desde todos los desarrollos presentados sobre las variantes de la MetaHeurística PSO. De modo que procede, primeramente, caracterizar el Espacio de Búsqueda en las que las formas X-FPSO serán aplicadas. Existen dos características básicas: **a)** Supone problemas de optimización combinatoria, dado el número de estados posibles que deben ser considerados y **b)** La búsqueda de soluciones más satisfactorias no puede seguir reglas de mejoras sustentadas en una función gradiente. Esto significa que las mejores soluciones para la Clase de Problemas a tratar, se encuentran aisladas, de modo que no existen cambios pequeños en alguna dirección que puedan guiar a la MetaHeurística hacia una mejora segura en la solución. A partir de esta caracterización, se ha podido concluir en la siguiente síntesis para la relación {X forma FPSO – Principios de IG Mejor Satisfechos}: **X1:** para cada uno de los cinco principios, requiere de un ajuste específico de parámetros, de modo que se la presenta como base, y no será empleada en el dominio de la HiperHeurística; **X2:** Habilidad específica en la Estabilidad; **X3:** Habilidad específica para la Proximidad y Estabilidad; **X4:** Habilidad general en los cinco principios, fundamentalmente en Calidad y Adaptación; constituye, por ello, la X forma de inicio en instancias tempranas de decisión y **X5:** Habilidad específica en Diversidad de Respuesta.

3.5 Función de Aprendizaje/Selección de la HY X-PSO CBR Aproximada mediante una Red Neuronal de Retropropagación

La Red Neuronal de Retropropagación (RNR) (SIMPSON, 1992), constituye un aproximador universal de funciones. Básicamente, tiene un diseño cuya unidad es la neurona artificial, que se agrupa en *capas*. Cada neurona recibe estímulos de entrada, provenientes desde aquellas que integran la capa precedente, y genera un resultado o estímulo de salida, el que, a su vez, constituirá una entrada para cada neurona de la *capa* siguiente. Existen dos capas características, que se denominan de Entrada y de Salida. En la Entrada, se presentan los patrones o estímulos a los que la Red debe responder, y en la Salida, los patrones *de* respuesta esperados. El número de neuronas que integran estas capas, queda definido por la cantidad de entradas y de salidas que son requeridas en la función.

Entre ambas capas, existe un *número* indeterminado de capas Ocultas, integradas cada una por un número también indeterminado de neuronas Ocultas. Dichos números, son resultado del diseño más adecuado. La conexión entre neuronas de capas contiguas, recibe el nombre de sinapsis, y su valor es referido como peso sináptico. El proceso de entrenamiento de una RNR, consiste en determinar los valores de los pesos sinápticos tales que minimicen el error global existente, entre la respuesta esperada y la efectivamente obtenida. Los detalles de diseño, se reservan para la segunda parte del presente trabajo, que consiste en la aplicación de la HiperHeurística propuesta, *para* definir el número de Estados de cada Etapa en la Optimización Dinámica Posibilística, asociada a la Planificación de Mediano/Corto Plazo de un SDEE. Sí corresponde, aquí, relacionar los elementos desarrollados y características descritas en la subsección anterior, con el diseño de la Red Neuronal: **1ro)** Los Patrones de Entrada serán números binarios, que representan, en grupos de 5, la habilidad o carencia de ella, que una solución obtenida a partir de cierta forma X-FPSO, cuya aplicación está en curso, exhibe en cierta instancia de decisión, para satisfacer los cinco Principios de IG; **2do)** Los Patrones de Salida, serán también número binarios, específicamente 7, en dos grupos: el primero, de cinco salidas, refiere, en el mismo orden que en las entradas, los Principios de IG que deberían mejorarse (1 \equiv mejora, 0 \equiv no es evaluado); el segundo, de dos salidas, representa la recomendación sobre cuál de las cuatro X-FPSO formas es conveniente aplicar. Desde este enfoque, la HiperHeurística es Basada en Razonamiento: el entrenamiento de la red, supone el acopio de una enorme cantidad de simulaciones, para componer los patrones de entrada y salida, de modo que la HiperHeurística “razone” sobre sus resultados y defina la aplicación más apta de la forma X-FPSO.

4. CONCLUSIONES

Complementariamente a los desarrollos y conceptos vertidos en esta primera parte del trabajo, se establecen tres conclusiones fundamentales; las dos últimas serán mejor observadas en su segunda parte, relacionada con la aplicación: **1ro)** Se ha presentado una exhaustiva revisión bibliográfica, a efectos de aportar, según los métodos de aprendizaje en el estado del arte, la novedosa HiperHeurística HY X-FPSO CBR propuesta, desde la experiencia de los autores en desarrollos MultiObjetivo sobre variantes de la MetaHeurística PSO. Con las mismas se compone su dominio; **2da)** El Algoritmo HiperHeurístico propuesto, utiliza técnicas compatibles con el paradigma referido en la introducción y **3ra)** Un problema de las características planteadas, no podría ser resuelto sino por una metodología de este tipo. La HiperHeurística propuesta, se plantea como una interesante estrategia para toda Clase de Problemas, que reúnan las condiciones generales aquí descritas.

REFERENCIAS

- BAI R. Y KENDALL G. (2006): "An Investigation of Automated Planograms Using a Simulated Annealing Based Hyper-Heuristic". GECCO'06, July 2006, Seattle, Washington, USA.2005.
- BELLMAN R. Y ZADEH L. (1970): "Decision-Making in a Fuzzy Environment". Management Science, 17, pp. 141-164.
- BENI G., WANG J. (1989): "Swarm Intelligence in Cellular Robotic Systems". Proceedings NATO Advanced Workshop on Robots and Biological Systems, June 1989, Tuscany, Italy.
- BLACK P. (2009): "Dictionary Of Algorithms and Data Structures". National Institute of Standards and Technology. Versión obtenida el 1/06/2009. <http://www.nist.gov/index.html>.
- BURKE E., KENDALL G., O'BRIEN R., REDRUP D., SOUBEIGA E. (2003-c): "An Ant Algorithm Hyperheuristic". The Fifth Metaheuristics International Conference. Kyoto, Japan, August 2003.
- BURKE E., KENDALL G., SOUBEIGA E. (2003-a): "A Tabu-Search Hyperheuristic for Timetabling and Rostering". Journal of Heuristics, No 9, pp. 451-470.
- BURKE E., LANDA SILVA J., SOUBEIGA E. (2003-b): "Hyperheuristic Approaches for MultiObjective Optimisation". The Fifth Metaheuristics International Conference, pp. 052-{1-6}. Kyoto, Japan, August 2003.
- BURKE E., MACCARTHY B., PETROVIC S., QU R. (2002): "Knowledge Discovery in a HyperHeuristic for Course Timetabling Using Case-Based Reasoning". Proceedings of the Fourth International Conference on the Practice and Theory of Automated Timetabling (PATAT'02), pp. 90-103, Ghent, Belgium, August 2002.

- COELLO C., SALAZAR M. (2002): "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization". Proceedings of the 2002 Congress on Evolutionary Computation-CEC02, Vol. 2, pp. 1051-1056. Honolulu, Hawaii USA.
- COWLING P., KENDALL G., E. SOUBEIGA E. (2000): *A Hyperheuristic Approach to Scheduling a Sales Summit*. Selected Papers of the Third International Conference on Practice and Theory of Automated Timetabling, PATAT 2000. LNCS, pp. 176-190, Konstanz, Germany, August 2000. Springer-Verlag.
- COWLING P., KENDALL G., E. SOUBEIGA E. (2001): "A Parameter-Free Hyperheuristic for Scheduling a Sales Summit". In proceedings of 4th International Conference, Porto Portugal, July 2001, pp. 127-131.
- COWLING P., KENDALL G., E. SOUBEIGA E. (2002-a): *A Tool for Rapid Prototyping in Scheduling and Optimisation*. Second European Conference on Evolutionary Computing for Combinatorial Optimisation, EvoCop 2002. LNCS, pp. 1-10, Kinsale, Ireland, April 2002. Springer-Verlag.
- COWLING P., KENDALL G., E. SOUBEIGA E. (2002-b): *Hyperheuristic: A Robust Optimisation Method Applied to Nurse Scheduling*. Parallel Problem Solving from Nature VII, PPSN 2002. LNCS, pp. 851-860, Granada, Spain, September 2002, Springer-Verlag.
- DÍAZ A., GLOVER F., GHAZIRI H., GONZALEZ J. (1996): *Optimización Heurística y Redes Neuronales*. Paraninfo. Madrid, España.
- DORNDORF U. Y PESCH E. (1995): "Evolution Based Learning in a Job Shop Scheduling Environment". Computers and Operations Research, 22, pp. 25–40.
- FANG H. Y ROSS P. (1994): "A Promising Hybrid GA/Heuristic Approach for Open-Shop Scheduling Problems". Eleventh European Conference on Artificial Intelligence, pp.197–202.
- GLOVER F. (1989): "Tabu Search: Part I". In: *ORSA Journal on Computing*, vol. 1, pp. 190-206.
- KENNEDY J., EBERHART R. (1995): "Particle Swarm Optimization", IEEE International Conference on Neural Networks. IEEE Service Center, Piscataway, NJ. Pert, Australia.
- KITANO H. (1990): "Designing Neural Networks using Genetic Algorithms with Graph Generation System". Complex Systems, 4, pp. 461–476.
- LAVOIE, M. (1992). *Foundations of PostKeynesian Economic Analysis*. Edward Elgar Publishing. University of Ottawa, Canada.
- NORENKOW I. Y GOODMAN E. (1997): "Solving Scheduling Problems via Evolutionary Methods for rule Sequence Optimization". Second World Conference on Soft Computing, pp. 61–64.
- OSMAN I. Y KELLY J. (eds.) (1996): *Meta-Heuristics: Theory and Applications*. Ed. Kluwer Academic. Boston, USA.

- PETROVIC S., QU R. (2002): "Case-Based Reasoning as a Heuristic Selector in a Hyper-Heuristic for Course Timetabling Problems". Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies, Proceedings of KES'02, Vol 82, pp. 336-340, Crema, Italy, September 2002.
- ROSS P., SCHULENBURG S., MARÍN-BLÁZQUEZ J., HART E. (2002): "Hyper-heuristics: Learning to Combine Simple Heuristics in Bin-Packing Problems". Genetic And Evolutionary Computation Conference (GECCO 2002), New York, July 2002.
- SCHWEICKARDT G. Y MIRANDA V. (2010): "Metaheurística FEPSO Aplicada a Problemas de Optimización Combinatoria: Balance de Fases en Sistemas de Distribución Eléctrica". Revista Ciencia Docencia y Tecnología, Universidad Nacional de Entre Ríos, Nro. 40, pp. 133-163.
- SCHWEICKARDT G., MIRANDA V., GIMENEZ J. (2010): "Dos Enfoques Metaheurísticos para resolver Problemas de Optimización Combinatoria Multicriterio: Fuzzy Evolutionary Particle Swarm Optimization con Topología Estrella Global/Individual (FEPSO GIST) y Fuzzy Simulated Annealing (FSA)". Anales del Encuentro XXIII ENDIO - XXI EPIO - II ERABIO, Tandil, Argentina.
- SIMPSON P. (1992): *Foundations of Neural Networks*. Artificial Neural Networks. IEEE Press. New York.
- SMITH J. (2002): *Parallel problem solving from nature VII*. Granada, Spain. Springer-Verlag.
- TCHEPRASOV V. Y GOODMAN E. (1996): "A Genetic Algorithm to generate a Pro-active Scheduler for a Printed Circuit Board Assembly". First International Conference on Evolutionary Computation and its Applications, pp. 232–244.