

## **Implementación de la metaheurística FEPSO GIST mediante procesamiento paralelo: Aplicación al problema de balance de fases en sistemas de distribución en baja tensión**

*Gustavo Schweickardt, Carlos Casanova*

Consejo Nacional de Investigaciones Científicas y Técnicas - CONICET, Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay (UTN-FRCU), Ing. Pereira 676, Concepción del Uruguay, Argentina, 3260.

Autor para correspondencia: gustavoschweickardt@conicet.gov.ar

Fecha de recepción: 23 de septiembre de 2014 - Fecha de aceptación: 19 de octubre de 2014

### **RESUMEN**

El presente trabajo describe el diseño e implementación de la Metaheurística FEPSO GIST (Fuzzy Particle Swarm Optimization with Global-Individual Star Topology) bajo un paradigma de computación paralela, empleando la plataforma MPI (Message Passing Interface), y aplicada a la solución del problema de Balance de Cargas en Sistemas de Distribución de Energía Eléctrica en Baja Tensión (PBC SDEE BT). En primer lugar, se plantea el Problema PBC SDEE BT multiobjetivo, carente de solución por métodos de optimización clásica. En segundo lugar, se presenta el marco de trabajo sobre el cual las heurísticas poseen un amplio grado de aplicación, y donde la computación paralela se yergue como una aliada invaluable al intentar, al igual que los métodos heurísticos, acelerar la respuesta de algoritmos que requieren de un poder computacional superior al paradigma secuencial. Seguidamente, se describe el diseño del algoritmo FEPSO GIST, la alternativa elegida para su implementación en MPI, y los detalles a tener en cuenta para una mejor ejecución. Por último, Se presenta su aplicación en un PBC SDEE BT real.

Palabras clave: Metaheurísticas, enjambre de partículas, optimización multiobjetivo, computación paralela, balance de cargas, sistemas de distribución.

### **ABSTRACT**

This paper describes the design and implementation of the FEPSO GIST (Fuzzy Particle Swarm Optimization with Global-Individual Star Topology) Metaheuristic, under a paradigm of parallel computing platform using the MPI (Message Passing Interface). The model is applied to the solution of Multiobjective Unbalanced Load Problem in Low Voltage Electric Distribution Systems (ULP LV EDS). First, the ULP LV EDS is presented, which has no solution by classical optimization methods. Then, the framework over the heuristics methods has a wide range of application and the parallel computation is the main approach to accelerate the results of algorithms that require a more powerful computing process respect of sequential paradigm, is described. Finally, the FEPSO GIST Metaheuristic design, in this framework, is presented, and its implementation MIP is applied on a real ULP LV EDS.

Keywords: Metaheuristics, particle swarm, multiobjective optimization, parallel computation, unbalance load problem, distribution systems.

## 1. INTRODUCCIÓN

### 1.1. El problema de balance de fases

Los Algoritmos Metaheurísticos son ampliamente reconocidos como uno de los más exitosos enfoques para la solución de Problemas de Optimización Combinatoria. Este es el caso del Problema del Balance de Cargas en un Sistema de Distribución de Energía Eléctrica en Baja Tensión (SDEE) desde un planteo Multiobjetivo. Este problema no tiene solución realista aplicando técnicas derivadas de la optimización clásica (Schweickardt & Miranda, 2010). El SDEE BT parte de Centros de Transformación de Media Tensión a Baja Tensión (CT MT/BT), en los que se emplazan uno o más transformadores, que presentan cierto número de salidas en BT. Dicho número aumenta con la potencia instalada. Desde cada salida (en un transformador para un CT tipo plataforma, de 315 (kVA), se tienen 4-6 salidas), parte un alimentador que, a su vez, tiene ramales en derivación, llamados secundarios (los cuales pueden tener derivaciones terciarias, y así siguiendo), conformándose una red independiente cuya topología de operación es radial. El sistema es trifásico, pero las cargas conectadas en los diferentes nodos, resultan, en su mayoría, monofásicas. Esto le confiere un carácter, si bien simétrico, desbalanceado. Significa que el sistema de BT, opera con cierto grado de desbalance en sus tres fases.

Sin embargo, en la medida que tal desbalance exhiba un grado más pronunciado, se presentan, al menos tres problemas relacionados con las pérdidas, la calidad del suministro y la confiabilidad del sistema:

- Existe un aumento en las pérdidas técnicas (potencia y energía) del sistema. En segmentos urbanos/suburbanos, para sistemas de BT que utilizan 0,38 (kV) de tensión nominal de suministro, el nivel de pérdidas técnicas de potencia se sitúa entre un 5 a 10%. El último valor se observa, mayormente, en zonas carenciadas, debido al tipo constructivo de red que se emplea (frecuentemente, conductores desnudos referidos como convencionales);
- La calidad del producto técnico tensión de suministro, controlada por la regulación, disminuye. La tolerancia en tensión para redes como las descritas en el ítem anterior, está entre un 5 a 7% respecto de la tensión de suministro nominal;
- El desbalance de intensidades, para sistemas con neutro puesto a tierra, genera una corriente de secuencia cero,  $I^{0j}$  (suma de las intensidades de las tres fases, que circulará por tierra) de un valor tal que puede accionar las protecciones contra cortocircuitos. Particularmente, el monofásico/fase a tierra (el más frecuente: un 80% de las fallas son de este tipo). El sistema pierde así confiabilidad, puesto que las protecciones interpretan los fuertes desbalances como una señal de cortocircuito, interrumpiendo el servicio. Esto adquiere mayor relevancia en el subsistema de MT: las cargas para el mismo resultan ser los CT MT/BT (sin considerar los grandes usuarios en MT, que son pocos, relativamente). Los desbalances en BT, se reflejan, entonces, “aguas arriba” como cargas desbalanceadas en MT. En este subsistema existen protecciones accionadas por componente homopolar mayor que cierto nivel de disparo, requiriéndose especial cuidado con su ajuste.

Entonces, el problema de optimización en el grado de desbalance de fases, es definido como la búsqueda de aquella configuración de conexiones de las cargas a las fases [R, S, T] en el sistema, tal que propenda a evitar, en el mayor nivel posible, los inconvenientes descritos en a), b) y c). Como cada carga del sistema tiene tres opciones de conexión, se trata de un problema combinatorio, en rigor dado por las variaciones con repetición de las 3 fases tomadas de a  $nC$  cargas. El espacio de búsqueda tendrá, entonces,  $NE = 3^{nC}$  estados. Para un sistema como el considerado en la simulación del presente trabajo, con  $nC = 115$  cargas, se tiene un espacio de búsqueda de  $NE = 3^{115}$  estados.

Este problema ha sido resuelto por los autores, siguiendo los desarrollos y aportes del primer autor, en términos del enfoque metaheurístico FEPSO GIST. También ha sido resuelto mediante otras Metaheurísticas de las denominadas formas X-FPSO (Multiobjetivo) propuestas en (Schweickardt & Miranda, 2010). La que mejor performance ha tenido, marcadamente, ha resultado ser la FEPSO GIST, motivo por el cual su algoritmo, detallado más abajo, fue seleccionado para su implementación bajo computación paralela mediante la plataforma de desarrollo MPI, en lenguaje C, corriendo bajo el Sistema Operativo Linux.

## 1.2. Motivación de la computación paralela

El mundo del software ciertamente ha revolucionado la industria. Hoy en día contamos con una amplia gama de aparatos electrónicos de uso cotidiano gobernados por software: microondas, cajeros automáticos, teléfonos inteligentes; aunque también de uso específico, como equipamiento médico, radares y otros sistemas usados para la guerra. Todo el tiempo se pretenden y requieren nuevas aplicaciones, con *mayores costos computacionales*, para cubrir nuevas necesidades. Sin embargo, el rendimiento de las computadoras dentro del paradigma secuencial clásico está comenzando a saturarse: existen límites físicos asociados al incremento en la frecuencia del reloj. Es por eso que los equipos multiprocesadores y/o multinúcleo han surgido, para combatir estos límites físicos, ya no intentando extender ese límite, sino agregando más unidades de cómputo, para que en conjunto brinden una mejor performance que el procesamiento secuencial. Se torna, por tanto, primordial, aprovechar al máximo este esquema de hardware mediante un nuevo concepto de software: la Computación Paralela.

La plataforma MPI (Message Passing Interface) o Interfaz de Paso de Mensajes, constituye un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores (núcleos). Su principal característica es que no precisa de memoria compartida, por lo que es muy importante en la programación de sistemas distribuidos.

Los elementos principales que intervienen en el paso de mensajes son el proceso que envía, el que recibe y el mensaje. Dependiendo de si el proceso que envía el mensaje espera a que el mensaje sea recibido, se puede hablar de paso de mensajes *síncrono* o *asíncrono*. En el paso de mensajes *asíncrono*, el proceso que envía, no espera a que el mensaje sea recibido, y continúa su ejecución, siendo posible que vuelva a generar un nuevo mensaje y enviarlo antes de que se haya recibido el anterior. Por este motivo se suelen emplear buzones, en los que se almacenan los mensajes a espera de que un proceso los reciba. Generalmente empleando este sistema, el proceso que envía mensajes sólo se bloquea o detiene, cuando finaliza su ejecución, o si el buzón está lleno. En el paso de mensajes *síncrono*, el proceso que envía el mensaje espera a que un proceso lo reciba para continuar su ejecución. Por esto esta técnica se suele llamar *encuentro*, o *rendez vous*.

Si bien la Interfaz de Paso de Mensajes (conocida ampliamente como MPI, siglas en inglés de Message Passing Interface) es un protocolo de comunicación entre computadoras, es el estándar para la comunicación entre los nodos que ejecutan un programa en un sistema de memoria distribuida. De manera que tales nodos pueden ser procesadores o núcleos que integran el hardware de cómputo en una misma computadora, bajo un sistema operativo capaz de explotar esta cualidad eficientemente (tal como Linux). Las implementaciones en MPI consisten en un conjunto de bibliotecas de rutinas que pueden ser utilizadas en programas escritos en los lenguajes de programación C, C++, Fortran y Ada. La ventaja de MPI sobre otras bibliotecas de paso de mensajes, es que los programas que utilizan la biblioteca son portables (dado que MPI ha sido implementado para casi toda arquitectura de memoria distribuida), y muy rápidos, (porque cada implementación de la biblioteca ha sido optimizada para el hardware en la cual se ejecuta).

Como se dijo, el sistema operativo para soportar la plataforma MPI es, en este caso Linux, y el lenguaje de programación escogido para el implementar el diseño de la Metaheurística FEPSO GIST paralelizada, que resuelve el problema PBC SDEE BT, ha sido C (ANSI).

## 2. FUNDAMENTOS TEÓRICOS

### 2.1. La metaheurística EPSO GIST monoobjetivo

El primer enfoque PSO (Particle Swarm Optimization) u Optimización por Enjambre de Partículas, fue propuesto en (Kennedy & Eberhart, 1995). Este tipo de solución puede tratar con problemas de optimización Monoobjetivo. Se lo conoce como *PSO Canónico*, y muchas variantes o formas X-PSO, pueden ser consultadas en la referencia (Schweickardt & Miranda, 2010).

En el modelo *PSO Cánico* existe, en cada iteración hacia la solución buscada, un conjunto de alternativas (cuya factibilidad exhibe cierto grado de aptitud), las cuales son denominadas *partículas*. El conjunto es la población o *enjambre*. 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:

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

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

$$X^{[k+1]}_i = X^{[k]}_i + V^{[k+1]}_i \quad (2)$$

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

$$V^{[k+1]}_i = V^{[k]}_i + w_C x (r_1^{[k]}) x [b^{[k]}_i - X^{[k]}_i] + w_S x (r_2^{[k]}) x [b^{[k]}_G - X^{[k]}_i] \quad (3)$$

En (3), el primer término 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  $k$ .

Desde los diferentes aportes realizados atentos a la performance de las formas X-PSO Monoobjetivo, se encuentra la Metaheurística EPSO (Evolutionary Particle Swarm Optimization) u Optimización Evolucionaria por Enjambre de Partículas propuesta por (Miranda *et al.*, 2008). Constituye un algoritmo de optimización metaheurístico que integra los conceptos de Estrategias Evolutivas (Evolution Strategic) y Optimización de Enjambre de Partículas (Particle Swarm Optimization). Los autores proponen conferirle al PSO una capacidad auto-adaptativa, tal que permita a la metaheurística desarrollar un proceso de cambio de comportamiento, conforme resulte la evolución de las soluciones. La auto-adaptación, permitiría que, en cada iteración  $k$ , los parámetros que requieren de ajuste externo puedan adaptarse sin riesgo de una convergencia prematura a un óptimo local. El mecanismo del algoritmo EPSO se puede describir de la siguiente manera: para una iteración  $k$  del PSO, las partículas evolucionarán a lo largo de un cierto número de generaciones, según los siguientes operadores evolutivos:

- a) Replicación: cada partícula es replicada un número  $r$  de veces, generando partículas iguales a las existentes;
- b) Mutación: los parámetros estratégicos (constantes de inercia, cognitiva y social, esencialmente) son mutados (adoptan, en rigor, el carácter de parámetros con la iteración  $k$ );
- c) Evaluación: cada sucesor concebido a través de los pasos anteriores, es evaluado mediante la función de aptitud definida;

- d) Selección: mediante algún proceso (típicamente, el torneo estocástico), las mejores partículas sobreviven para formar la nueva generación, que se somete, en la iteración siguiente, a la nueva regla de movimiento.

La ecuación evolutiva para la regla del movimiento en el EPSO, se sintetiza en la velocidad de la partícula  $i$ -ésima, entendiendo que la iteración  $k$  se corresponde con una nueva generación:

$$\mathbf{V}^{[k+1]}_i = w_{iL}^{*[k]} \times (r_1^{[k]}) \times \mathbf{V}^{[k]}_i + w_{iC}^{*[k]} \times (r_2^{[k]}) \times [\mathbf{b}^{[k]}_i - \mathbf{X}^{[k]}_i] + w_{iS}^{*[k]} \times (r_3^{[k]}) \times [\mathbf{b}^{[k]*}_G - \mathbf{X}^{[k]}_i] \quad (4)$$

donde el superíndice (\*), significa que los parámetros son evolutivos, producto de la mutación.

La regla de mutación aplicable a las constantes  $w_{iL, C, S}$  tiene, como expresión general:

$$w_{iL, C, S}^{*[k+1]} = w_{iL, C, S}^{[k]} \times [1 + \sigma \times N(0, 1)] \quad (5)$$

en la cual:  $\sigma$  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:

$$\mathbf{b}^{[k+1]*}_G = \mathbf{b}^{[k]*}_G + w_{iN}^{*[k]} \times N(0, 1) \quad (6)$$

donde  $w_{iN}^{*[k]}$ , es un cuarto 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  $\mathbf{b}^{[k]*}_G$  actual, para la iteración  $k$ ). El nuevo parámetro, como se indica con el superíndice (\*), es también mutado según la regla (5). Se infiere que en la regla de movimiento modificada para el EPSO, el vector cooperación mutado no apuntará en la dirección de  $\mathbf{b}^{[k]}_G$ , sino de  $\mathbf{b}^{[k]*}_G$ .

Un aspecto muy importante que se define desde la formulación primigenia del PSO Canónico, es la Topología de Comunicación entre las partículas del enjambre. Esto es, la forma en que cada una de ellas recibe la información del Óptimo Global (OG) y, en algunas formas X-PSO, de los Óptimos Individuales de todas o de ciertas partículas (OI<sub>*i*</sub>). Normalmente, tal topología es Síncrona y Global, de modo que todas las partículas reciben a la vez la información del OG, tal y como una suerte de estrella, cuyo centro es la partícula que, en cierta iteración del algoritmo solidario a la Metaheurística, y desde el cual se “irradia” la información del OG al resto. Por tal motivo, esta Topología de Comunicación recibe el nombre de Estrella Global Determinística.

Una primera modificación introducida por Miranda *et al.* (2008) sobre la Metaheurística EPSO, con buenos resultados, propendiendo a la diversidad de posiciones que las partículas del enjambre adopten en el Espacio de Búsqueda en cada iteración del algoritmo, es la Topología de Comunicación Estrella Global Estocástica. Su concepto altera los términos de la comunicación de aquella posición o partícula (solución) que ha alcanzado en cierta iteración,  $k$ , el Óptimo Global (OG),  $\mathbf{b}^{[k]*}_G$ , que es un vector de tantas dimensiones como lo establece el espacio de búsqueda, según se dijo. En lugar de que todas las partículas se dejen influenciar del mismo modo por la que detenta el OG, conforme se aprecia en la Ecuación Evolutiva de Movimiento EPSO, (4), que se corresponde con una topología Global, se asigna una probabilidad de que dicho OG afecte o no a las restantes partículas del enjambre y su valor es una cuantía de afectación entre 0 y 1.

Desde aquí, y también propendiendo a una mayor capacidad de autoajuste de parámetros y diversidad en las posiciones en las partículas se introduce un nuevo modelo de comunicación síncrono, denominado por el primer autor del presente artículo Topología de Comunicación Estrella Estocástica Global/Individual. Esto genera una nueva X-PSO forma, que recibe el nombre de EPSO GIST, Metaheurística aún aplicable en el dominio Monoobjetivo. Se sostiene la idea de Estrella Estocástica, en la vecindad del OG, definida por probabilidades de comunicación no nulas. Pero se incorpora la probabilidad, que hasta el momento es nula, de que también las partículas incidan sobre otras informándoles de su posición. Los Óptimos Individuales (OI) de las partículas del enjambre, tienen ahora cierta probabilidad, que puede ser nula, de incidir sobre el resto. Se introduce, además, un Factor de Decaimiento,  $\delta(k)$ . Este factor multiplicativo,  $\delta(k)$ , va haciendo “decaer” la influencia de los óptimos individuales, OI, con probabilidades distintas de 0, desde cada partícula del enjambre hacia el resto, conforme aumenta el número de iteraciones. Con esto se logra una capacidad auto-adaptativa adicional en la Metaheurística para Espacios de Búsqueda como el que exhibe el PBC

SDEE BT: las partículas exhiben una mayor diversidad de posiciones en iteraciones tempranas, para luego dejarse influir por el OG evolutivo, a medida que el decaimiento de las incidencias estocásticas de los OI, se va sucediendo conforme aumenta el número de iteraciones. La Ecuación Evolutiva de Movimiento para la Metaheurística EPSO GIST, viene dada por la expresión:

$$\mathbf{V}^{[k+1]}_i = w_{iI}^{*[k]} \times (r_1^{[k]}) \times \mathbf{V}^{[k]}_i + w_{iC}^{*[k]} \times (r_2^{[k]}) \times [\mathbf{b}^{[k]}_i - \mathbf{X}^{[k]}_i] + w_{iS}^{*[k]} \times (r_3^{[k]}) \times [\mathbf{b}^{[k]*}_G - \mathbf{X}^{[k]}_i] \times \alpha OG^{[k]}_i + \sum_{j=1, j \neq i}^n (\delta(k) \times w_{iC}^{*[k]} \times [\mathbf{b}^{[k]}_j - \mathbf{X}^{[k]}_i] \times \alpha OI^{[k]}_{ij}) \quad (7)$$

donde:  $\alpha OG^{[k]}_i$  es la probabilidad de que el Óptimo Global afecte al movimiento de la partícula  $i$ , en la iteración  $k$  y  $\alpha OI^{[k]}_{ij}$  es la probabilidad de que el Óptimo Individual de la partícula  $j$ , afecte al movimiento de la partícula  $i, j \neq i$ .

## 2.2. La metaheurística FEPSO GIST multiobjetivo

La manera de extender la Metaheurística EPSO GIST Monoobjetivo, al dominio Multiobjetivo que se ha propuesto, consiste en el empleo de Conjuntos Difusos solidarios a cada uno de los Objetivos y/o Restricciones del problema, (Schweickardt & Miranda, 2010), al introducir la toma de decisión en ambientes difusos. Para ello se propone trabajar con el concepto de Variables de Apartamiento, solidarias a cada conjunto difuso Objetivo/Restricción, resultando los mismo mapeados por igual en el Conjunto Difuso de Decisión, sobre el cual la Metaheurística buscará la solución más satisfactoria, operando sobre una combinación de las funciones de pertenencia correspondientes. Por ello el nombre FEPSO GIST (Fuzzy EPSO GIST) para referir la nueva Metaheurística, inherentemente, como Multiobjetivo.

Una Variable de Apartamiento,  $u$ , solidaria a cierta variable Objetivo/Restricción del problema (de Estado),  $v$ , con un valor aceptable o de referencia,  $vRef$ , adopta la forma:

$$u = (v - vRef)/vRef \quad (8)$$

Para el PBC SDEE BT, puesto que los cuatro objetivos deben ser minimizados, puede también adoptarse una formulación diferente, preservando la idea sintetizada en (8). En los objetivos Pérdidas de Potencia ( $perd$ ), Componente Homopolar circulante por el neutro del SDEE BT considerado ( $I^{(0)}$ ), y Número de Cambios de Fase de las cargas ( $NCf$ ), podrían definirse valores de referencia admisible ( $vRef$ ) y mínimos ( $vMin$ ), quedando la variable de apartamiento solidaria, definida, como:

$$u = (v - vMin)/(vRef - vMin) \quad (9)$$

Como se observa en (8) y (9),  $u = \mu(v)$  es la función de pertenencia de un conjunto difuso lineal, elegido así por simplicidad, en principio. Esta estructura seguirá siendo válida para el Índice de Tensión Fuera de Tolerancia para los nodos del Sistema, con algunos desarrollos auxiliares que se presentarán más abajo, puesto que es el único objetivo que debe ser maximizado. En este diseño, se proponen simular Optimizaciones  $PSO$  Mono-Objetivo para obtener los valores  $vMin$ , y apelar a los valores de contexto, operativo o regulatorio, para obtener  $vRef$ . Esto es: *minimizar pérdidas, minimizar componente homopolar referida a la salida, y maximizar el Índice de Tensión fuera de Tolerancia*, respectiva e individualmente. Y luego pueden, entonces, establecerse las referencias pertinentes.

De modo que, para cada uno de los tres primeros objetivos, se tendrán los siguientes conjuntos difusos, definidos mediante sus funciones de pertenencia expresadas en forma condicional:

### 1.- Pérdidas ( $perd$ )

$$\mu(perd) = 1; \text{ si } perd \leq Minperd \quad (10)$$

$$\mu(perd) = [(Refperd - perd) / Refperd]^{p(perd)}; \text{ si } Minperd \leq perd \leq Refperd \quad (11)$$

$$\mu(perd) = 0; \text{ si } perd > Refperd \quad (12)$$

$p(perd)$  es un ponderador exponencial, que *contrae* al conjunto difuso, dándole mayor importancia al objetivo en la *confluencia* (Bellman & Zadeh, 1970) u operación entre los valores de las funciones de pertenencia,  $\mu(v)$ , si  $p(perd) > 1$ , o lo *dilata* (menor importancia del objetivo), si  $p(perd) < 1$ .

2.- Componente Homopolar ( $I^{(0)}$ )

$$\mu(I^{(0)}) = 1; \text{ si } I^{(0)} \leq \text{MinCH} \quad (13)$$

$$\mu(I^{(0)}) = [(RefCH - I^{(0)}) / RefCH]^{p(I^{(0)})}; \text{ si } \text{MinCH} \leq I^{(0)} \leq RefCH \quad (14)$$

$$\mu(I^{(0)}) = 0; \text{ si } I^{(0)} > RefCH \quad (15)$$

Lo dicho para  $p(perd)$  vale también para  $p(I^{(0)})$ .

3.- Número de Cambios de Fase (NCf)

Se calcula, como se explicó mediante la Metaheurística PSO Monoobjetivo:

$$RefNCf_{Max} = MAX \{NCf_{PSOMinperd}; NCf_{PSOMinCH}; NCf_{PSOMaxITS}\} \quad (16)$$

Para poder obtener un *límite mínimo admisible*, una posibilidad es obtener el mínimo de los valores de (16), al cual se le sustrae un cierto valor  $NCf_0$ , fijado externamente:

$$RefNCf_{Min} = MIN \{NCf_{PSOMinperd}; NCf_{PSOMinCH}; NCf_{PSOMaxITS}\} - NCf_0 \quad (17)$$

Luego, la expresión para la función de pertenencia correspondiente al conjunto difuso  $NCf$ , es la siguiente:

$$\mu(NCf) = 1; \text{ si } \forall t \text{ } NCf \leq RefNCf_{Min} \quad (18)$$

$$\mu(NCf) = [(NCf - RefNCf_{Min}) / (RefNCf_{Max} - RefNCf_{Min})]^{p(NCf)}; \text{ si } RefNCf_{Min} \leq NCf \leq RefNCf_{Max} \quad (19)$$

$$\mu(NCf) = 0; \text{ si } NCf > RefNCf_{Max} \quad (20)$$

Lo dicho para  $p(perd)$  vale también para  $p(NCf)$ .

4.- Índice de Tensión Fuera de Tolerancia (ITS)

- Se fijan: un *límite inaceptable* de tensión en los nodos terminales,  $u_{Inad}$ , y un *límite aceptable* en los mismos,  $u_{Ad}$ . Los empleados en la simulación son:  $u_{Inad} = 0.92$  [pu] y  $u_{Ad} = 0.95$  [pu];
- Se definen los valores recíprocos de las tensiones límites especificadas en **a)**:  $v1 = 1/u_{Ad}$ , y  $v2 = 1/u_{Inad}$ ;
- Dado un valor de tensión,  $u$ , en cierto nodo terminal,  $nt$ , se calcula su recíproco,  $vft = 1/u$ .

Luego, la expresión condicional para la *función de pertenencia* correspondiente al conjunto difuso *tensión fuera de tolerancia*, de variable  $vft$ , evaluada en el nodo terminal  $nt$ , resulta:

$$\mu(vft) = 1; \text{ si } vft \leq v1 \quad (21)$$

$$\mu(vft) = [(vtf - v1) / (v2 - v1)]^{p(vft)}; \text{ si } v1 \leq vft \leq v2 \quad (22)$$

$$\mu(vft) = 0; \text{ si } vft > v2 \quad (23)$$

Lo dicho para  $p(perd)$  vale también para  $p(vft)$ .

Finalmente, como Índice de Tensión fuera de Tolerancia,  $ITS$ , se propone la media geométrica de todos los valores  $\mu_{nt}(vft)$  en el conjunto de los nodos terminales,  $\{NT\}$ :

$$\mu(U_{tft}) = \sqrt[n_{NT}]{\prod_{nt=1}^{n_{NT}} \mu_{nt}(vft)} \quad (24)$$

Para componer las funciones de pertenencia en cada iteración, se emplea como operador de confluencia la t-norma Producto de Einstein (Schweickardt & Miranda, 2010), definida como:

$$PET: (x,y) \rightarrow \frac{x \times y}{2 - (x + y - x \times y)} \quad (25)$$

la cual en este caso sigue la siguiente secuencia de construcción:

$$TPE1 = \frac{\mu(perd) \times \mu(|I^{[0]}|)}{2 - (\mu(perd) + \mu(|I^{[0]}|) - \mu(perd) \times \mu(|I^{[0]}|))} \quad (26)$$

$$TPE2 = \frac{TPI \times \mu(utf)}{2 - (TPI + \mu(utf)) - TPI \times \mu(utf)} \quad (27)$$

y, por último, siendo  $\mu_D$  la Decisión Maximizante:

$$\mu_D = \frac{TPE2 \times \mu(ncf)}{2 - (TPE2 + \mu(ncf)) - TPE2 \times \mu(ncf)} \quad (28)$$

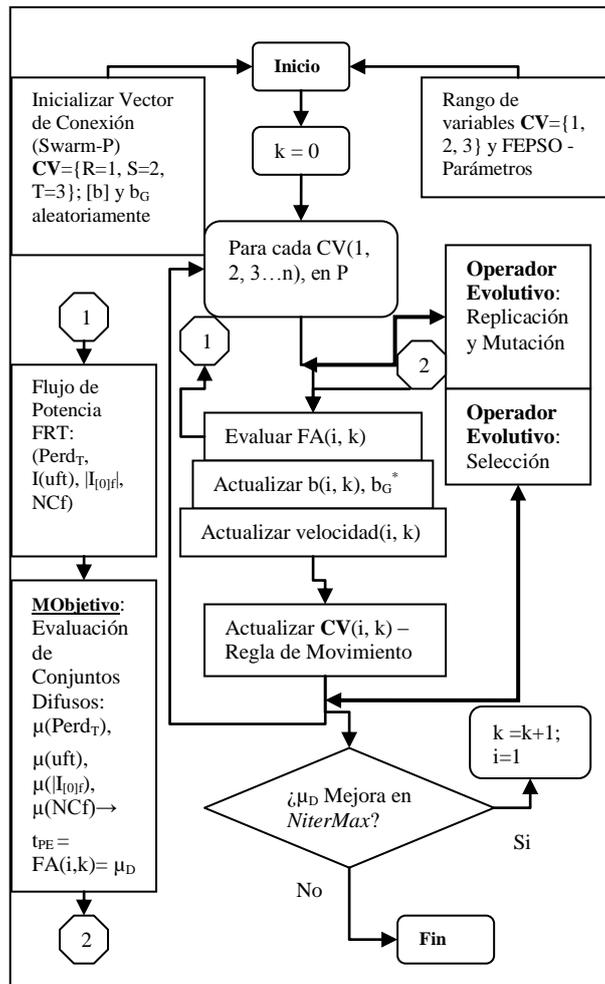


Figura 1. Diagrama de Flujo Secuencial de la FEPSO GIST.

### 2.3. El diseño FEPSO GIST bajo computación paralela

En la Fig. 1, se observa el *diseño secuencial*, mediante un diagrama de flujo, de la Metaheurística FEPSO GIST. En el diseño paralelo, el planteo más simple, seguido aquí, consiste en asignar a cada uno de los procesadores o núcleos, un proceso secuencial idéntico, pero que trate con un número de partículas resultante de dividir el número poblacional,  $NP$ , por el número de núcleos empleados,  $Nn$ . De modo que cada procesador ejecutará el diagrama secuencia de la Fig. 1, con  $np = NP/Nn$  partículas. Para obtener los mejores resultados, es entonces necesario especificar a MPI la cantidad de núcleos que tiene cada computadora. Esto se realiza mediante la especificación de un archivo llamado *machinefile*. En este caso, el archivo resultante constó de una sola línea: localhost: 4. Esta línea se corresponde con el nombre del equipo, junto a la cantidad de núcleos del mismo. 4 resultó el número

de núcleos con mejor performance para este problema, debido a que a mayor cantidad de núcleos, mayor costo computacional de comunicación se tendrá entre procesos secuenciales. Posiblemente, con otros diseños, desde el vastísimo campo de investigación que se abre aquí, se tengan mejores resultados aprovechando un mayor número de núcleos. La implementación de la Metaheurística FEPSO GIST bajo este esquema es bastante directa. Sólo serán destacados los detalles de las operaciones concernientes a MPI. Estos detalles son: (a) la comunicación de la mejor aptitud, y (b) la comunicación de la posición de la partícula más apta. Para comunicar la aptitud máxima se utilizó la función MPI\_Allreduce, con operación MPI\_MAXLOC. Para la ejecución de esta instrucción, cada proceso aporta la aptitud de su mejor partícula, junto a su rango o número de proceso, en una estructura MPI\_FLOAT\_INT. Cada partícula recibe el máximo de dichas aptitudes, junto al rango o número de proceso que se encuentra en posesión de dicha partícula.

Para componer las funciones de pertenencia en cada iteración, se emplea como operador de confluencia la t-norma Producto de Einstein (Schweickardt & Miranda, 2010), definida como:

Para compartir la posición de la mejor partícula se utiliza la función MPI\_Bcast. Cada proceso aporta su mejor del enjambre local para que en él se mapee el mejor valor de todos los procesos. El valor de la raíz del MPI\_Bcast es el calculado mediante MPI\_Allreduce, según lo especificado anteriormente. Las ventajas descriptas para esta nueva topología GIST, son adoptadas en cada proceso individual, por núcleo.

### 3. SIMULACIÓN Y RESULTADOS

En la Fig. 2 se presenta el diagrama trifilar de una salida real de un CT MT/BT en la ciudad de San Carlos de Bariloche, Argentina. Las cargas son monofásicas. Respecto de la configuración óptima de las cargas logradas, es idéntica a la resultante mediante la aplicación de otras X-FEPSO formas, como las presentadas en las referencia (Schweickardt & Miranda, 2010), la cuales puede ser consultadas. Tampoco son diferentes los resultados alcanzados para cada objetivo planteado. Lo que sí resulta muy diferente, como era de esperar, es la drástica reducción en el tiempo de ejecución y convergencia de la Metaheurística FEPSO GIST bajo Computación Paralela. En el caso de un único núcleo o procesador (FEPSO GIST Secuencial) se alcanzaron los siguientes resultados: *Número de Partículas de la Población: 200; Pérdidas [kW]: 7.21; |I<sup>0</sup>| [A]: 0.4;  $\mu(U_{ft})$ : 0,27; NCf: 59; Tiempo de Ejecución: 47 min.*

Bajo Computación Paralela, 4 núcleos con 50 partículas por núcleo/proceso, se alcanzaron los mismos resultados, *pero en un tiempo de ejecución de 9 min.* Por supuesto, la eficiencia de la codificación en Lenguaje de Programación C, aportó rapidez en la ejecución.

### 4. CONCLUSIONES

Ha sido introducida una nueva Metaheurística introduciendo una propuesta en la Topología de Comunicación, Estrella Estocástica Global/Individual, dando origen a la Metaheurística Multiobjetivo FEPSO GIST. La misma exhibe excelentes resultados en términos de diversidad y estabilidad de las soluciones, con el mejor tiempo de respuesta para el PBC SDEE BT, bajo un paradigma secuencial.

Por tal motivo, ha sido seleccionada para un diseño bajo un paradigma de Computación Paralela, empleando la plataforma MPI o Interfaz de Paso de Mensajes, bajo el Sistema Operativo Linux, y utilizando el Lenguaje de Programación ANSI C. Los resultados son contundentes, en términos de la drástica reducción el tiempo de ejecución.

Por último, este paradigma amerita otras clases de diseños por caso, explotando las características de lo Principios de la Inteligencia de Grupo y su relación con esta Clase de Problemas, para definir mejores criterios de asignación de tareas a los procesos paralelizados.

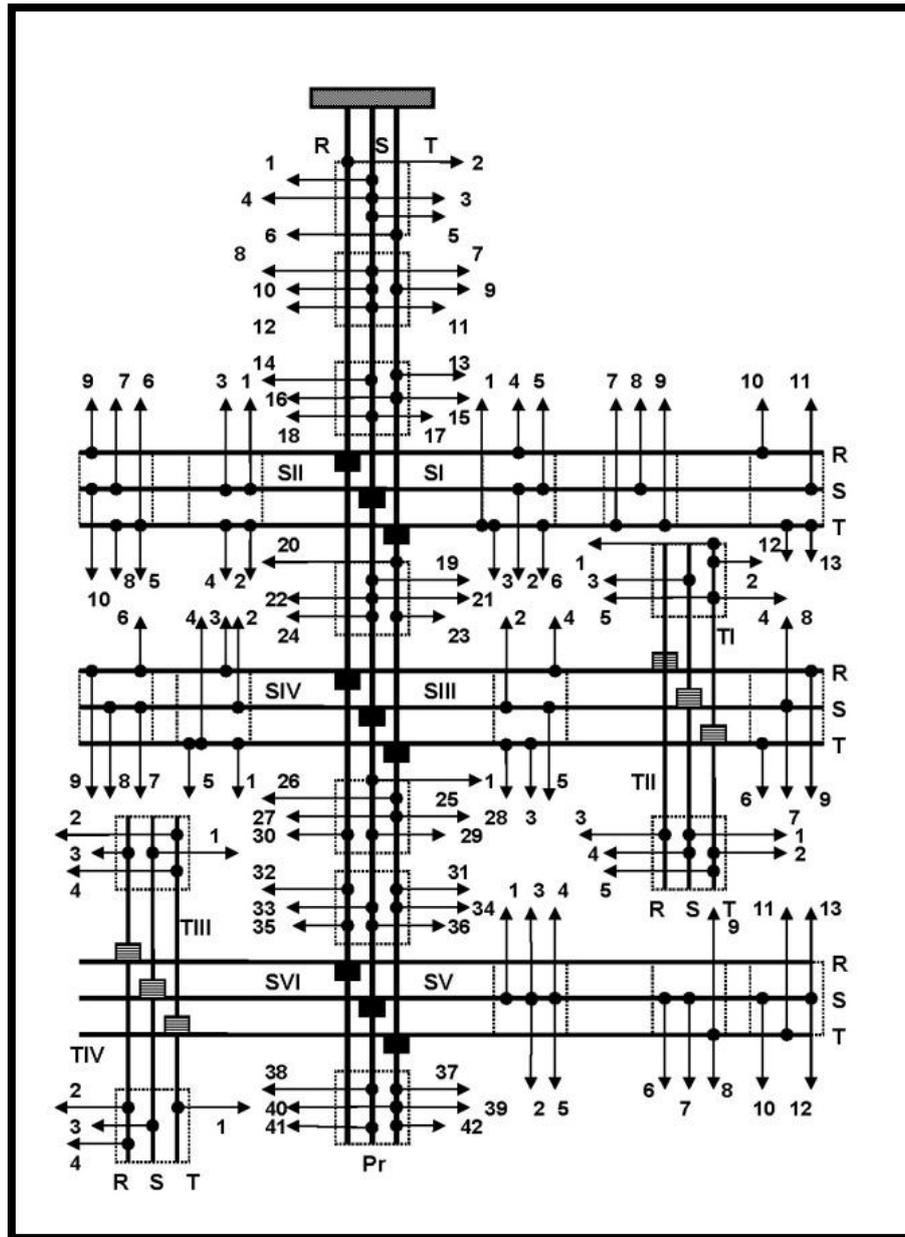


Figura 2. Diagrama Trifilar de la salida de un CT MT/BT considerado.

## REFERENCIAS

- Bellman, R.E., L.A. Zadeh, 1970. Decision making in a fuzzy environment. *Management Science*, 17(4), B141-B164.
- Kennedy, J., R.C. Eberhart, 1995. Particle Swarm Optimization. *IEEE International Conference on Neural Networks*. Pert, Australia, 6 pp.
- Miranda, V., H. Keko, A. Duque Jaramillo, 2008. EPSO: Evolutionary Particle Swarms. In: *Advances in Evolutionary Computing for System Design*. Jain, L., V. Palade, D. Srinivasan (Eds.), Springer Series in Computational Intelligence, 66, 139-168.
- Schweickardt, G., V. Miranda, 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*, XXI(40), 133-163.