

# Implementación del algoritmo VNS-DEEPSO para el despacho de energía en redes distribuidas inteligentes

## Implementation of the VNS-DEEPSO algorithm for the energy dispatch in smart distributed grid

DOI: <http://dx.doi.org/10.17981/ingecuc.15.1.2019.13>


Artículo de Investigación Científica. Fecha de Recepción:17/09/2018. Fecha de Aceptación:05/04/2019.

**Pedro Julián García-Guarín** 

Universidad Nacional de Colombia. Bogotá, (Colombia)  
pjgarrciag@unal.edu.co

**Julián Cantor-López** 


Universidad Nacional de Colombia. Bogotá, (Colombia)  
jrcantorl@unal.edu.co

**Camilo Cortés-Guerrero** 

Universidad Nacional de Colombia. Bogotá, (Colombia)  
caacortesgu@unal.edu.co

**María Alejandra Guzmán-Pardo** 

Universidad Nacional de Colombia. Bogotá, (Colombia)  
maguzmanp@unal.edu.co

**Sergio Rivera-Rodríguez** 

Universidad Nacional de Colombia. Bogotá, (Colombia)  
srriverar@unal.edu.co

Para citar este artículo:

Pedro Julián García-Guarín; Julián Cantor-López; Camilo Cortés-Guerrero; María Alejandra Guzmán-Pardo; Sergio Rivera-Rodríguez "Implementación del algoritmo VNS-DEEPSO para el despacho de energía en redes distribuidas inteligentes", *INGE CUC*, vol. 15, no. 1, pp. 142-154, 2019. DOI: <http://doi.org/10.17981/ingecuc.15.1.2019.13>

### Resumen

**Introducción**– Las redes eléctricas tradicionales están migrando a nuevas configuraciones de redes inteligentes, que traen retos operacionales y de planeación. Con miras a avanzar en estos retos se propone resolver un problema de optimización usando programación en elementos de redes distribuidas inteligentes.

**Objetivo**– El problema de optimización consiste en administrar el despacho energético de una red inteligente para optimizar los recursos disponibles, considerando la incertidumbre de energías renovables, viajes planeados de vehículos eléctricos, el pronóstico de carga y los precios del mercado.

**Metodología**– Se propuso utilizar un ensamble entre dos métodos heurísticos. El algoritmo VNS (Variable Neighborhood Search) y el DEEPSO (Differential Evolutionary Particle Swarm).

**Resultados**– El algoritmo VNS-DEEPSO fue evaluado en una competencia de "Smart Grids" con otros algoritmos con un valor de 18.21, siendo 7% mejor que el segundo algoritmo clasificado en la competencia.

**Conclusiones**– El algoritmo VNS-DEEPSO fue ganador entre 9 algoritmos metaheurísticos que solucionaron el problema, este problema tenía un mayor incremento de dificultad debida a la incertidumbre generada por factores ambientales, pronóstico de carga, viajes en vehículos eléctricos y el mercado de precios. Acorde a los resultados, el algoritmo VNS-DEEPSO demostró ser el más eficiente en minimizar los costos operacionales y maximizar los ingresos de la red inteligente.

**Palabras clave**– Algoritmos Heurísticos; Energía Renovable; Optimización; Smart Grid; Vehículos Eléctricos.

### Abstract

**Introduction**– Traditional electric networks are migrating to new configurations of intelligent networks, which bring operational and planning challenges. In order to advance in these challenges, an optimization problem is proposed to solve in the programming of intelligent network elements.

**Objective**– The optimization problem consists of managing the energy dispatch of an intelligent network to optimize the available resources, considering the uncertainty of renewable energies, planned trips of electric vehicles, cargo forecast and market prices.

**Methodology**– It was proposed to use an assembly between two heuristic methods. The VNS algorithm (Variable Neighborhood Search) and the DEEPSO (Differential Evolutionary Particle Swarm).

**Results**– The value obtained by the VNS-DEEPSO algorithm was 18.21, being 7% better than the second algorithm classified in the competition.

**Conclusions**– The VNS-DEEPSO algorithm was the winner among 9 metaheuristic algorithms that solved the problem. This problem has a greater increase in difficulty due to the uncertainty generated by weather conditions, load forecast, planned EV's trips, and market prices. According to the results, the VNS-DEEPSO algorithm proved to be the most efficient in minimizing operational costs and maximizing the revenues of the intelligent network.

**Keywords**– Heuristic Algorithms; Renewable energy; Optimization; Intelligent Network; Electric Vehicles.



I. INTRODUCCIÓN

La preservación ambiental y el desarrollo sostenible se han convertido en preocupaciones a nivel mundial dadas las manifestaciones de cambio climático y el continuo crecimiento en la demanda de energía. A medida que las ciudades cambian al uso de tecnologías más avanzadas, los consumos de electricidad aumentan a niveles que no pueden ser manejables y que no pueden dejar de atenderse. Las redes inteligentes ofrecen una respuesta al cambio a tecnologías más sostenibles como la generación distribuida y las microrredes [1].

El comienzo de las “Smart Grids” junto con la creciente incursión de la generación distribuida ha traído un nuevo nivel de complejidad en la planeación de la operación de sistemas de potencia. Dicha complejidad reside en la consideración de variables estocásticas en la formulación matemática de problemas de optimización, asociados con la creciente penetración de las energías renovables [2].

El despacho óptimo de energía, una vez organizado en una matriz de elementos como la energía solar fotovoltaica y la demanda de carga, tienen como problema que la función objetivo que realiza la planeación, la cual comprende un modelo que incluye funciones probabilísticas. Esta función produce valores de unidades de costos y cantidades de generación de potencia [3].

En Sudáfrica, las fuentes de energía renovables contribuyen sobre el 10% del total del consumo de energía. Mantener el balance entre suplir energía y la demanda de carga de los usuarios en tiempo real, llega a ser más difícil para suplir los recursos de generación y la demanda de carga. Ahora, se considera pertinente incorporar nuevas herramientas para lograr la confiabilidad bajo gran incertidumbre de la red [3].

Dada la importancia del manejo del recurso energético, diversas formulaciones matemáticas han sido ampliamente propuestas para resolver el

problema. Sin embargo, debido al nivel de complejidad, las formulaciones tradicionales que fueron diseñadas para escenarios completamente diferentes no pueden resolver el problema de manera eficiente [2]. Es en esas situaciones donde los enfoques tradicionales fallan, pero se ha demostrado que los algoritmos de alto nivel, también conocidos como metaheurísticas, han demostrado ser una herramienta muy poderosa para resolver este tipo de problemas.

En este sentido, en el Congreso Mundial de Inteligencia Computacional del 2018 (que agrupa los Congresos de Redes Neuronales; de Computación Evolucionaria, y de Sistemas Difusos) que tuvo lugar en Río de Janeiro (Brasil), se mostraron los resultados de la solución de un problema de optimización que se formuló en el primer semestre de 2018. Este problema debía ser resuelto con técnicas de alto nivel en donde se buscaba operar el despacho energético de una red inteligente, al mínimo costo, considerando incertidumbre en fuentes renovables, vehículos eléctricos, pronóstico de carga y precios del mercado [4].

II. FORMULACIÓN DEL PROBLEMA

En el ámbito del congreso de la IEEE: World Congress on Computational Intelligence - 2018 (WCCI-2018), el GECAD (Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development) del Politécnico de Porto, en colaboración con la Universidad de Delft, propusieron la competencia para la optimización de un problema donde la gestión de los recursos energéticos diarios en “Smart Grids” se encuentra centralizada y en entornos con incertidumbre (Fig. 1), [4]. Los lectores e investigadores interesados pueden ingresar al sitio de la conferencia, y bajar los códigos de la competencia para intentar mejorar los resultados presentados en este artículo.

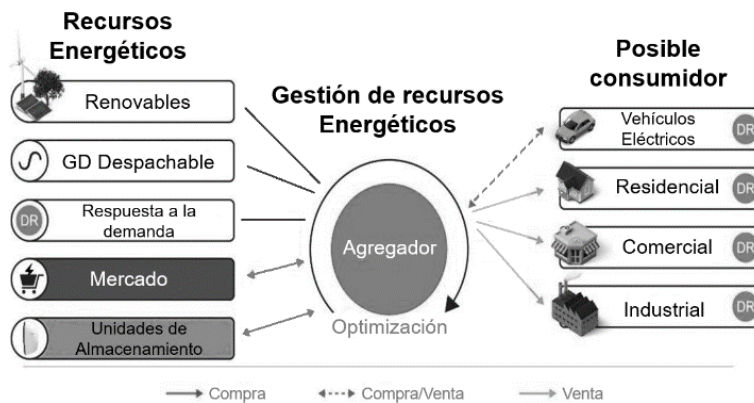


Fig. 1. Descripción general del problema de gestión energética. Fuente: imagen adaptada de [4].

Este problema considera un intermediario distribuidor de energía (de ahora en adelante agregador), el cual tiene las siguientes roles:

- Satisface las necesidades energéticas de recursos distribuidos y del mercado eléctrico.
- Minimiza los costos operacionales y obtiene ingresos por la venta de energía en mercados eléctricos disponibles.
- Utiliza activos propios como Sistemas de Almacenamiento de Energía para abastecer la carga de la demanda.
- Establece contratos bilaterales de energía con quienes buscan suministro de electricidad, por ejemplo, clientes residenciales e industriales.
- No obtiene beneficios del suministro de energía a cargas fijas y carga de vehículos eléctricos.
- Realiza la programación para el día siguiente, considera las condiciones climáticas (para predecir la generación renovable), la demanda de la carga, los viajes de los vehículos eléctricos y los precios del mercado [2]-[4]. Sin embargo, esto puede traer graves consecuencias en la operación de la red cuando las suposiciones no son altamente precisas.

En consecuencia, se necesita que el agregador cuente con la capacidad necesaria para encontrar soluciones robustas e inherentes a algunos parámetros del entorno. La idea principal es que el algoritmo propuesto realice la programación de los recursos energéticos dedicados de las 24 horas del día siguiente.

Por lo tanto, el agregador debe encontrar soluciones que proporcionen no sólo un valor óptimo (o cercano al óptimo) de los costos operacionales, sino que esas soluciones deben ser tan poco sensibles como sea posible a las variaciones de los parámetros inciertos. La incertidumbre para este tipo de problemas se clasifica en cuatro categorías: ruido, robustez, función “fitness” aproximada y función “fitness” variable en el tiempo [5]. Esta competencia se ubicó en la categoría de robustez en la que las variables de diseño, o parámetros ambientales en este caso, están sujetas a perturbaciones después de que se haya determinado la solución óptima.

Para incorporar incertidumbre en los parámetros, los diseñadores de la competencia utilizaron simulaciones de Monte Carlo para generar un gran número de posibles escenarios basados en funciones de distribución de probabilidad de los errores de pronóstico obtenidos de datos históricos.

Un gran número de escenarios aumenta la precisión del modelo, pero viene con un costo de cálculo asociado con una gran cantidad de variaciones en los parámetros. Debido a esto, se utilizó una técnica de reducción para mantener una pequeña cantidad de escenarios a la vez que se mantienen las principales características estadísticas de los escenarios iniciales.

#### A. Estructura del Algoritmo

En la Fig. 2 se muestra la estructura utilizada por los organizadores de la competencia, donde los competidores debían utilizar una metaheurística para resolver el problema.

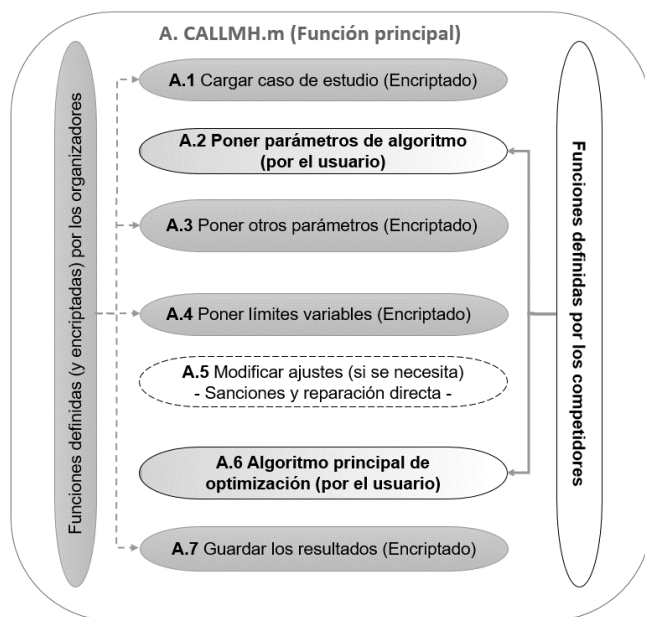


Fig. 2. Estructura general de la simulación. Fuente: imagen adaptada de [4].

El modelo se implementó en *Matlab*© 2016 de 64-bit y contiene una estructura de diferentes scripts con objetivos específicos en la simulación. En la Fig. 2 los rótulos indican que esas rutinas (*scripts*) fueron proporcionadas por los organizadores. Los competidores debían implementar dos programas:

1. Un programa para ajustar los parámetros requeridos por el algoritmo (A.2).
2. Un programa para la implementación del método de solución propuesto (A.6).

### B. Codificación

La estructura de la solución propuesta dependerá de la metaheurística utilizada, por ejemplo, para una Evolución Diferencial (DE) se tendrán individuos, en una solución tipo enjambre (PSO) se tendrán partículas, o en un Algoritmo Genético se tendrán genotipos. La solución adoptada en la competencia tiene la forma que se muestra en la Fig. 3. Donde se presentan los siguientes grupos de variables (1) Potencia activa, (2) Generadores binarios, (3) y (4) Carga y descarga de EVs y ESS, (5) Respuesta a la demanda y (6) Mercado.

TABLA 1. REPRESENTACIÓN DE LA SOLUCIÓN.

Grupo de variable	Tipo	Dimensiones de estudio
(1) Potencia Activa	Continua	7 (6DGs+1 suplementos)
(2) Generadores binarios	Binario	7 (6DGs+1 suplementos)
(3) Carga/Descarga EVS	Continua	34 Evs
(4) Carga/Descarga ESS	Continua	2 ESS
(5) Respuesta a la demanda	Continua	90 Cargas
(6) Mercado		2 Mercados
Número de variables por cada período		142 por cada período
Dimensión total para 24 períodos		D = 142 * 24 = 3408

Fuente: imagen adaptada de [4].

Cada solución se codifica como un vector conformado por 6 grupos de variables que se repiten secuencialmente a lo largo de los 24 períodos (horas) de optimización. En la representación vectorial todas las variables, sin incluir el grupo (2), las variables son continuas con límites que coinciden con los límites de potencia o capacidad de las variables asociadas.

El grupo (2) representa variables binarias que se utilizan para indicar si un generador está conectado (valor '1') o desconectado (valor '0'). Las variables binarias también pueden presentar un valor continuo ya que la función "fitness" corrige internamente su valor usando una operación simple de redondeo.

El grupo (1) pertenece a variables de generación distribuida (DG). Es importante notar que en este grupo se incluye generación de energía fotovoltaica

que no puede controlarse, por lo que incluso cuando forma parte de la solución vectorial, tendrá un valor específico, y por lo tanto inalterable, dependiendo del escenario considerado.

### C. Formulación matemática del problema

#### 1) Función "fitness" e incertidumbre.

La función "fitness"  $f'$  considera el objetivo  $Z$  del agregador más la suma de las penalidades encontradas durante la evaluación de las soluciones:

$$f'(\vec{X}) = Z + \rho \sum_{i=1}^{N_C} \max[0, g_i] \quad (1)$$

donde  $\vec{X}$  es una solución que sigue la estructura mostrada en la tabla 1. En este caso,  $g_i$  es el valor de la  $i$ -ésima construcción (igualdad o desigualdad) y  $\rho$  es un factor de penalización configurable que por lo general se considera grande.

De igual forma, en algunos parámetros se consideró incertidumbre para modificar el valor de  $f'$  acorde con diferentes escenarios generados por la simulación de Monte Carlo. El valor de  $f'$  se modifica con perturbaciones de la siguiente manera:

$$F_s(\vec{X}) = f'(\vec{X} + \delta_s) \quad (2)$$

donde  $\delta_s$  es la perturbación de variables y parámetros en el escenario  $s$ , y  $F_s(\vec{X})$  es el valor objetivo asociado al muestreo  $s$  de Monte Carlo. Por lo tanto, un valor promedio esperado para una solución dada sobre el conjunto de escenarios considerados se puede calcular como:

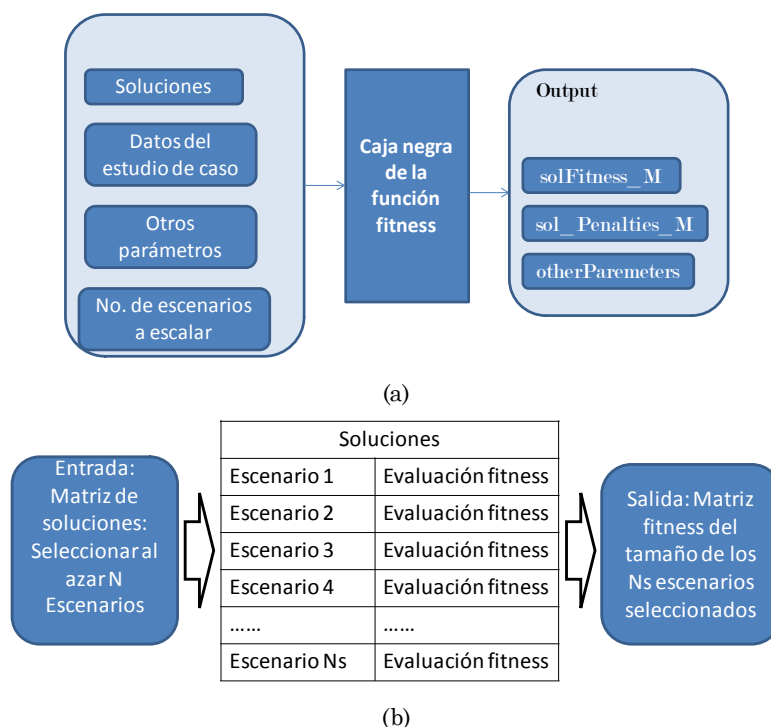
$$\mu_{FS}(\vec{X}) = \frac{1}{N_s} \cdot \sum_{s=1}^{N_s} f'(\vec{X} + \delta_s) \quad (3)$$

Del mismo modo, la desviación estándar de una solución en el conjunto de escenarios se puede calcular de la siguiente manera.

$$\sigma_{FS}(\vec{X}) = \sqrt{\frac{1}{N_s} \cdot \sum_{s=1}^{N_s} (f'(\vec{X} + \delta_s) - \mu_{FS}(\vec{X}))^2} \quad (4)$$

Las ecuaciones (3) y (4) dependen del número de escenarios considerados en la evaluación. Como se muestra a continuación, la función  $f'$  en el proceso de optimización recibe como parámetro la cantidad de escenarios que el competidor desea evaluar. Para la evaluación final de la competencia, las soluciones son evaluadas con un número total de 100 escenarios.

Los organizadores diseñaron esta función como una caja negra que se representa en la Fig. 3(a).  $f'$  es una función encriptada que recibe como argumento de entrada una matriz con las soluciones, la información del estudio de caso, algunos parámetros adicionales y el número de escenarios que el competidor desea evaluar, para la competencia, se consideraron un máximo de 100 escenarios.



**Fig. 3.** Función “fitness”: (a) Bloque de caja negra, (b) Funcionamiento interno.  
Fuente : Imagen adaptada de [4].

Como argumento de salida, la función devuelve una matriz con los valores de condición física de toda la población en un subconjunto de escenarios seleccionados al azar.

La Fig. 3 (b) muestra el funcionamiento interno de  $f'$ , que selecciona aleatoriamente  $N_{evals}$  escenarios ( $N_{evals}$  es un parámetro especificado por el usuario) de los  $N_s$  disponibles. Observe en esta figura que el número real de evaluaciones de la función depende del tamaño de la población a evaluar y de la cantidad de escenarios que el competidor desea considerar cada vez que se llama la función. El número de evaluaciones de funciones es, por lo tanto:

$$NFE = N_{sol} * N_{evals} \quad (5)$$

Los organizadores de la competencia restringieron el algoritmo de tal forma que un máximo 50,000 evaluaciones de funciones fueron permitidas.

## 2) Suposiciones del problema de programación de energía.

A continuación, se formulan los supuestos necesarios a la hora de solucionar el problema:

- a. El agregador minimiza los costos operativos mientras maximiza sus ganancias (costos menos ingreso).
- b. Los vehículos eléctricos se pueden controlar continuamente (entre 0 y la tasa de carga máxima).
- c. La misma suposición se aplica al principio V2G (Vehicle-to-grid) (entre 0 y la tasa de descarga máxima).
- d. Las baterías estacionarias o los sistemas de almacenamiento de energía se pueden controlar continuamente de forma similar a los EVs/V2G.
- e. Se supone que la función de costo de las unidades DG es lineal.
- f. Se supone que el agregador de energía puede presentar ofertas y solicitudes al mercado de electricidad.
- g. Los mercados en los que participa el agregador tienen diferentes límites para las ofertas y demandas.
- h. Se consideran dos mercados correspondientes a mercados mayoristas y locales.
- i. Se generan 5000 escenarios reducidos a 100 escenarios para simular la incertidumbre de los viajes de vehículos eléctricos, generación de energía fotovoltaica, variaciones de carga y precios de mercado.

### 3) Notas sobre la implementación del problema.

Finalmente, se describe a continuación algunos hechos que se tienen en cuenta a la hora de implementar una solución:

- Internamente en la función  $f'$ , se asume que las variables de carga y descarga para los Vehículos Eléctricos son iguales, de igual forma, se asumen valores positivos para la carga y valores negativos para la descarga con el fin de ahorrar memoria computacional.
- El mismo principio descrito anteriormente se aplicó a las variables de los Sistemas de Almacenamiento de Energía.
- Internamente, el valor de mercado es positivo para una oferta (venta) y negativo para una demanda (compra).
- Las variables binarias siempre se redondean internamente en la función objetivo.
- La reparación directa de la solución se usa en la función objetivo.
- La función  $f'$  selecciona internamente un subconjunto aleatorio de los 100 escenarios disponibles cada vez que la función es llamada.

### 4) Descripción del escenario

La competencia se basó en una microrred de 25 nodos que representaba un área residencial con 6

Generadores Distribuidos (5 unidades despachables y 1 generador fotovoltaico), 1 proveedor externo, 2 Sistemas de Almacenamiento de Energía, 34 Vehículos Eléctricos, y 90 cargas controlables (donde se suponen que ellas tienen tiempos de respuesta [3]-[4]). Además, se considera que dos mercados (mayorista y local) están disponibles para compra y venta de energía. La tabla 2 describe los recursos disponibles, el precio del kWh se da en unidades monetarias (*u.m.*).

### 5) Incertidumbre en los escenarios

Los organizadores dispusieron un conjunto de 5000 escenarios para generación fotovoltaica, consumo de carga y variaciones de precio de mercado. Para la generación de incertidumbre en la generación fotovoltaica, se utilizó un error del 15%. En cuanto a la carga pronosticada y los precios de mercado, usaron errores de 10% y 20% respectivamente.

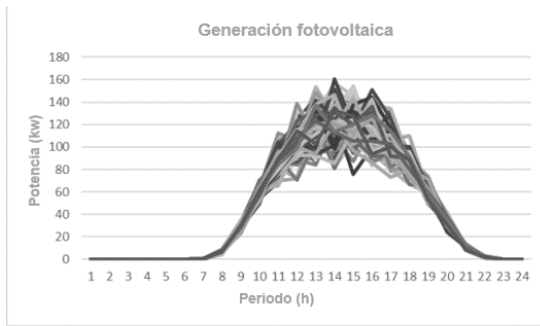
El número de escenarios se redujo a 100, utilizando técnicas de reducción especializadas. En cuanto a los viajes de los vehículos eléctricos, se generan aleatoriamente 100 diferentes pronósticos de programación para cada escenario. La Fig. 4, ilustra los escenarios generados para la generación fotovoltaica (a), pronóstico de carga (b), mercado mayorista (c) y mercado local (d).

TABLA 2. RESULTADOS RECONFIGURACIÓN SISTEMA DE DISTRIBUCIÓN DE MUESTRA

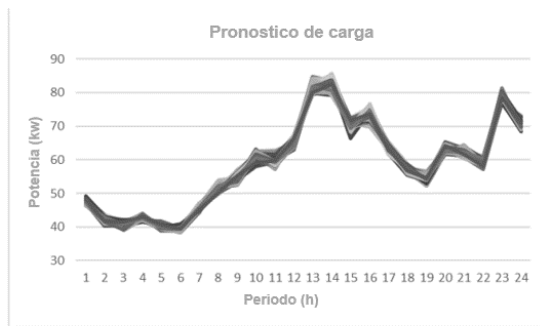
Recursos energéticos	Precios (u.m./kWh)	Capacidad (kW)	Cantidad
Generadores distribuidos despachables	0,07	10-100	5
Proveedores externos	0,074-0,16	0-150	1
Proveedores externos-Carga	-	0-16,6	
Sistemas de Almacenamiento de Energía-Descarga	0,03	0-16,6	2
Sistemas de Almacenamiento de Energía-Carga	-	0-111	
Vehículos Eléctricos-Descarga	0,06	0-111	34
Carga regulable de respuesta a la demanda	0,0375	4,06-8,95	90
<b>Pronostico (kW)</b>			
Fotovoltaico	-	0-106,81	1(17 agg)
Carga	-	35,82-83,39	90
<b>Limites (kW)</b>			
Mercado 1(Mayorista)	0,021-0,039	0-85	1
Mercado 2(Local)	0,021-0,039	0-40	1

Fuente: los autores basado en [4] (u.m. = unidades monetarias).

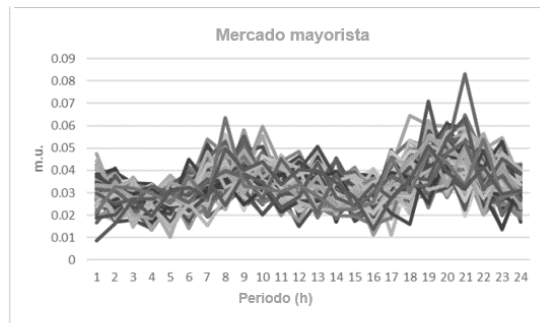
### 6) Reglas de la Competencia



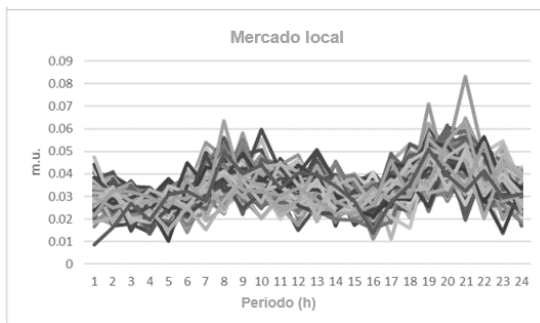
(a)



(b)



(c)



(d)

**Fig. 4.** Escenarios generados para la generación fotovoltaica (a), pronóstico de carga (b), mercado mayorista (c) y mercado local (d).

Fuente: Imagen adaptada de [4].

Las reglas establecidas por los organizadores de la competencia fueron las siguientes:

- Los participantes tenían que proponer e implementar algoritmos metaheurísticos para resolver el problema de gestión de recursos energéticos bajo incertidumbre.
- Los organizadores proporcionaron un programa, implementado en *MATLAB* © 2014b 64 bits, en el que los participantes podían correr sus algoritmos.
- Dado que los algoritmos propuestos podían tener distintos tamaños de población y ejecutarse para un número variable de iteraciones, se permitió un número máximo de 50000 evaluaciones de la función objetivo en cada prueba.
- Las propiedades de convergencia de los algoritmos no fueron un criterio de calificación.
- 20 ensayos independientes debían realizarse por cada participante.

### III. REVISIÓN LITERARIA Y METODOLOGÍA DE SOLUCIÓN: OPTIMIZACIÓN BASADA EN ENSAMBLE DE METAHEURÍSTICAS

las metaheurísticas son métodos de solución que organizan interacciones entre procedimientos de mejora local y estrategias de alto nivel para crear procesos capaces de escapar de óptimos locales y ejecutar búsquedas robustas en un espacio de solución [6].

Con el tiempo, las metaheurísticas han incluido procedimientos que emplee estrategias para superar la trampa de óptimos locales en espacios de solución complejos, especialmente aquellos procedimientos que utilizan una o más estructuras de vecindario como medio para definir movimientos admisibles para la transición de una solución a otra, o para construir o destruir soluciones en procesos constructivos y destructivos.

Dentro de las metaheurísticas, existe una variedad de estrategias, tales como los algoritmos genéticos, algoritmos miméticos, enjambre de partículas, búsqueda tabú, inteligencia de enjambre, búsqueda dispersa, re-encadenamiento de trayectoria, entre otras. En este estudio se escogieron las dos técnicas (sección 3.1 y 3.2) que daban mejores resultados en el problema formulado.

Para ello se utilizó la optimización basada en el ensamble de métodos heurísticos, que ha recibido recientemente gran atención como una técnica potencialmente poderosa para resolver las desventajas de las metaheurísticas [7]-[9]. Las publicaciones recientes han aumentado tanto el número de aplicaciones como la comprensión teórica del método. Sin embargo, todavía hay un amplio margen para un mayor desarrollo ya que la mayoría de la teoría se basa en la combinación secuencial de los algoritmos heurísticos. En este estudio se utilizó la técnica de Ensemble Optimization (EO) con una estrategia de evolución natural ya bien definida conocida como Mutación Gaussiana [8].

La fortaleza de la solución está en el método de gradiente aproximado conocido como técnica de optimización en ensamble [9]. La aproximación de gradiente en el EO se basa en una regresión lineal entre un conjunto de muestras de control y sus correspondientes valores de función objetivo. Las muestras de control se extraen de una distribución gaussiana multivariante con una matriz de covarianza definida por el usuario (constante) y una media conocida. Varias publicaciones [7]-[10] han demostrado que EO puede lograr buenos resultados de valor práctico en una variedad de diferentes modelos de optimización complejos [7]-[10].

A continuación, se introduce los algoritmos DEEPSO y VNS, utilizados para darle solución al problema de optimización de recursos de energía, planteado anteriormente.

#### A. *Differential Evolutionary Particle Swarm Optimization - DEEPSO.*

DEEPSO es una técnica híbrida de alto nivel construida a partir de diferentes metaheurísticas: “*Evolutionary Algorithms*” (EA), “*Particle Swarm Optimization*” (PSO) y “*Differential Evolution*” (DE). Estos métodos tienen como ventaja que impulsan a encontrar el óptimo en una dirección globalmente correcta. Al combinar estas técnicas se busca generar un algoritmo más robusto [10].

En EA, se utilizan mecanismos inspirados en la evolución biológica, como la reproducción, la mutación, la recombinación y la selección. Las soluciones candidatas al problema de optimización desempeñan el papel de los individuos en una población, y la función “*fitness*” determina la calidad de las soluciones. La evolución de la población se produce después de la aplicación repetida de las técnicas biológicas.

Al igual que el EA, el DE pertenece a la familia de los algoritmos evolutivos, sin embargo, es comparativamente más simple. El algoritmo DE tiene solo tres o cuatro parámetros operacionales, y se puede codificar en aproximadamente 20 líneas de pseudocódigo. A pesar de la aparente simplicidad del DE, los operadores evolutivos clave de interacción de mutación y recombinación están presentes y son efectivos. En particular, el DE tiene la ventaja de incorporar una forma relativamente simple y eficiente de mutación autoadaptable [11].

De otro lado, los algoritmos basados en enjambres surgieron como una poderosa familia de técnicas de optimización, inspiradas en el comportamiento colectivo de los animales sociales. En PSO, el conjunto de soluciones candidatas al problema de optimización se define como un enjambre de partículas que pueden fluir a través del espacio de parámetros definiendo trayectorias que son impulsadas por sus propios mejores rendimientos y los de sus vecinos [12].

La primera combinación entre estos algoritmos se dio entre EA y PSO. Este algoritmo se denominó EPSO: “*Evolutionary Algorithms Particle Swarm Optimization*”, fue presentado en 2002 e introdujo una forma de unir el poder exploratorio del PSO con la potencia auto-adaptativa del EA [10].

La idea detrás del algoritmo EPSO, consistió en proporcionar capacidad de adaptación al operador de recombinación. Para lograr esto, los parámetros en la partícula de velocidad (una parte de la estructura de PSO) están sujetos a mutación y selección con el fin de tratar de lograr una mayor tasa de progreso. Como resultado, en diversos trabajos se ha demostrado la calidad y fiabilidad del algoritmo, así como su buen rendimiento en una diversidad de dominios, como se muestra en las referencias [10], [11].

De igual forma, diversos autores han buscado la forma de combinar el algoritmo DE con otras técnicas de alto nivel, sin embargo, los resultados han sido indicativos y no deben tomarse como conclusiones apresuradas [10].

Finalmente, el DEEPSO se introdujo como una técnica metaheurística por primera vez en el 2013 [10]. En ese trabajo se demostró de manera deductiva que los resultados obtenidos eran de mejor calidad que si se aplicaban otras técnicas. El caso de estudio consistía en un sistema de potencia definido matemáticamente como un problema de optimización de costo fijo o de programación no lineal entera mixta.

En el problema, dado un conjunto de generadores y sus curvas de costos de generación, se debía definir cuales generadores tenían que ser apagados y cuales estar en servicio y en qué nivel de carga, con el fin de minimizar el costo general (costos de inicio más costos de operación).

#### B. *Variable Neighborhood Search – VNS*

Esta técnica cambió el enfoque con el que se resolvían los problemas. De manera tradicional los algoritmos estaban basados en búsqueda local, donde a partir de una solución inicial y por medio de una secuencia de cambios locales en cada iteración, el valor de la función objetivo era evaluada hasta que un óptimo local era encontrado [13].

Sin embargo, el algoritmo VNS basó su búsqueda en el cambio de vecindario. VNS no sigue una trayectoria, sino que explora vecindarios cada vez más distantes de la mejor solución alcanza en la respectiva iteración, y saltos de esta solución a una nueva si y sólo si se encuentra una mejora. De esta forma, se utilizan características favorables al alcanzar mayor exploración del espacio de búsqueda.

El algoritmo VNS presentado en esta competencia fue diseñado con algunas mejoras que permiten la solución de ecuaciones formadas a partir de ecuaciones algebraicas no lineales, y ecuaciones diferenciales en la que es imposible obtener su derivada con un modelo matemático exacto [14].



Una estrategia de búsqueda local para la optimización no lineal es capaz de encontrar un óptimo local de un problema que presenta muchas buenas soluciones de calidad. La estrategia consiste en extender la búsqueda local para realizar cambios sistemáticos en las vecindades, solucionando problemas complejos. El algoritmo explora vecindades distantes y actualiza la mejor solución. La estructura de funcionamiento del algoritmo tiene dos fases principales: inicio y repetición (ver Fig. 5) [14].

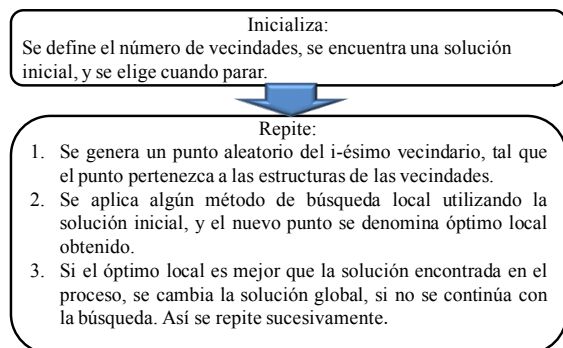


Fig. 5. Variable Neighborhood Search  
Fuente: [14].

En el paso 3 pueden ocurrir 3 situaciones: 1) Si el óptimo global es igual al óptimo local el algoritmo continúa en el próximo vecindario, 2) En el caso que el óptimo local encontrado es peor que el óptimo global, se continúa en el siguiente vecindario y 3) La tercera situación implica que el óptimo local es mejor que el óptimo global, entonces se actualiza el nuevo óptimo global [14].

Para mejorar el valor óptimo encontrado se continúa con otro método llamado método de coordenada cíclica que puede ser utilizado para optimizar funciones no diferenciable y no lineal. El algoritmo tiene dos fases, inicialización y paso principal [14].

En la inicialización se elige arbitrariamente un escalar mayor a cero para finalizar el algoritmo. También se elige el número de direcciones en los que se realizará la búsqueda de línea. En el segundo paso se permiten soluciones óptimas al problema a minimizar utilizando restricciones. Si la diferencia entre paso sucesivos se amplía entonces se para [14].

El método de búsqueda de línea de Fibonacci [3]-[4] se utiliza para encontrar el máximo de una función unimodal para reducir el intervalo de incertidumbre. Además, minimiza una función estrictamente cuasi convexa a un intervalo limitado. El método se aplica en dos pasos: 1) Inicialización y 2) Paso principal. 1) En la inicialización se escoge el tamaño de la incertidumbre y el número de cálculos y 2) Se evalúa la mejora de la función utilizando dos escalares, dependiendo el valor de los escalares se reduce el intervalo de incertidumbre [14].

#### IV. RESULTADOS

En total 9 equipos a nivel mundial enviaron propuestas de solución a la competencia. En la tabla 3 se muestra la posición final y la conformación de cada uno de los equipos, el nombre del algoritmo utilizado y el ranking index obtenido. El ranking index se calcula como la suma del promedio de 20 pruebas de la función "fitness"  $f$  más la desviación estándar de cada solución.

TABLA 3. RESULTADOS DE LA COMPETENCIA.

Posición	Equipo	Algoritmo utilizado	Ranking Index
1	Universidad Nacional de Colombia, ACCELOGIC y Khalifa University	VNS-DEEPSO	18,21
2	Charusat, Gujarat India	Enhanced Velocity Differential Evolutionary Particle Swarm Optimization-EVDEPSO	19,57
3	University of Porto	Chaotic Evolutionary Swarm Optimization	24,89
4	Universidad de Salamanca	Particle Swarm Optimization with Global Best Perturbation PSO-GBP	31,02
5	Charusat, Gujarat India	Improved_Chaoitic_Differential Evolution	34,52
6	University of Thessaloniki	Unified-PSO (UPSO)	36,28
7	Fundan University	Improved Differential Evolution	36,84
8	University, South Korea	Differential evolution with stochastic selection	52,83

Fuente: Autores

La potencia generada para cada generador se presenta en la Fig. 6. La generación de energía fotovoltaica se representa en la Fig. 7 por una curva que se mantiene en 0 durante las primeras 7 horas iniciando con una generación de energía desde los 6 kW con un suave incremento hasta los 126 kW, cae hasta un fondo alrededor de 87 kW a las 15 horas, seguido de un máximo alrededor de 116 kW y finaliza con un descenso escalonado hasta 0 kW.

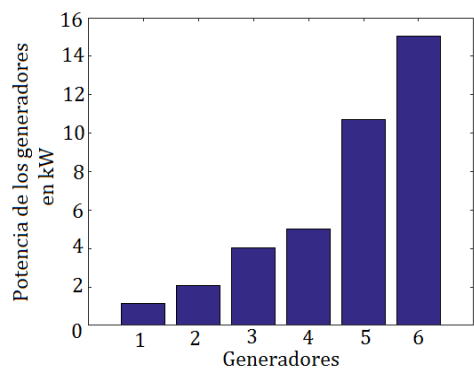


Fig. 6. Potencia de los generadores. Fuente: los autores.

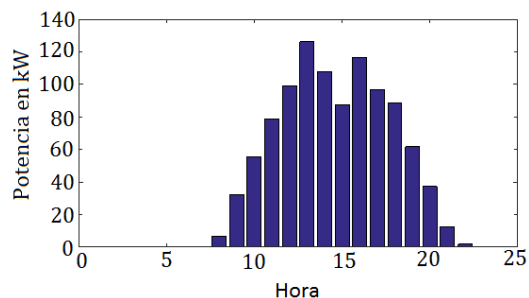


Fig. 7. Fuente de suministro externo de energía. Fuente: los autores

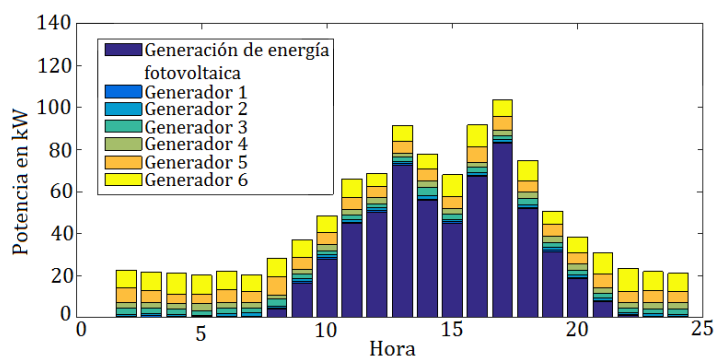


Fig. 8. Generación de potencia distribuida. Fuente: Autores

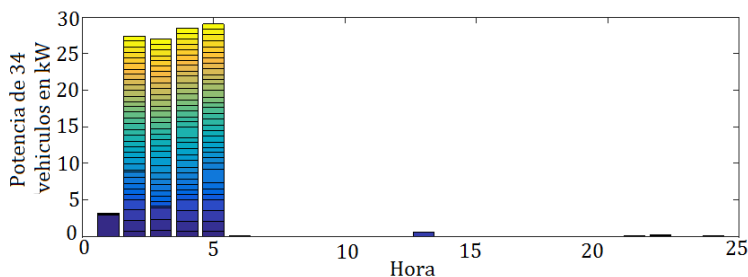


Fig. 9. Carga de 34 vehículos eléctricos. Fuente: Autores

La operación en conjunto de las 5 unidades de despacho, un generador de energía fotovoltaica y un proveedor externo muestran la curva de generación de energía para un día típico (ver Fig. 8), los generadores del 1 al 6 varían sus conexiones para generar la carga alrededor de los 20 kW que se mantiene con pequeñas variaciones durante las primeras 8 horas seguido de un incremento escalonado hasta la hora 14, después de bajar levemente hasta el fondo en la hora 15, llega a su máximo ascenso de 103,8 kW a la

hora 17 y cae levemente hasta un valor un poco sobre 20 kW en las 24 horas.

La Fig. 9 presenta el consumo de energía para 24 vehículos eléctricos, en la primera hora el consumo es de un poco más de 3 kW y continúa con un máximo consumo alrededor de 28 kW el cual se mantiene con pequeñas fluctuaciones entre la hora 2 y la hora 5, desde la hora 6 hasta la hora 24 el consumo de energía es poco significativo y se mantiene en valores cercanos a 0.

Las cargas mantienen consumos mínimos que están alrededor de cero, como se evidencia en la Fig. 10. Los consumos máximos se dan a las horas 5, 9 y 21 que están alrededor de respectivamente 0,009, 0,012 y 0,017 kW.

La Fig. 11, presenta la potencia vendida al por mayor y la potencia vendida en el mercado local, las ventas al por mayor tuvieron su máximo potencia a las 3 y 9 horas con valores de 84 y 76 kW, en las demás horas la potencia fluctuó entre los 56 kW. Mientras que en

el mercado local la compra de potencia se mantuvo estable con algunas fluctuaciones alrededor de los 26 kW.

La Fig. 12, ilustra la operación de dos sistemas de almacenamiento de energía que se descargan con una cumbre sobresaliente a la hora 2 de 3,22 y 1,06 kW para las unidades 1 y 2 respectivamente y alcanzan su pico máximo de descarga a la hora 4 de 11,99 y 4,36 kW para las mismas unidades. Los demás valores se mantienen oscilando cercanos a 0 para otras horas del día.

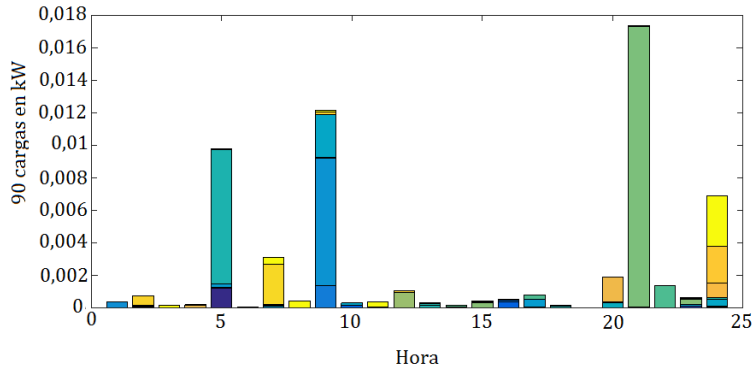


Fig. 10. Consumo de cargas.  
Fuente: Autores

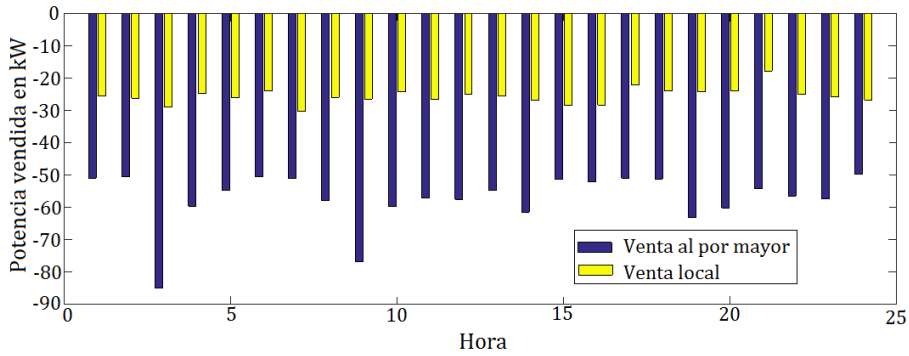


Fig. 11. Potencia vendida.  
Fuente: Autores

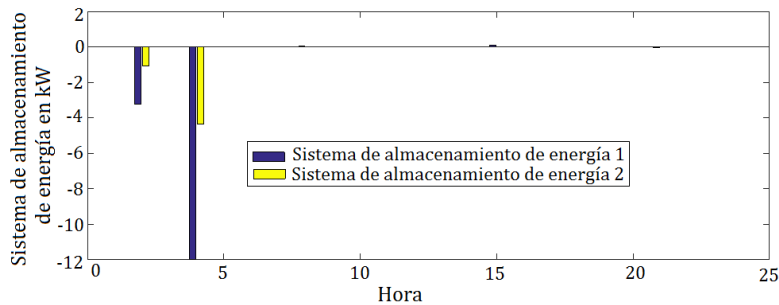


Fig. 12. Sistemas de almacenamiento de energía.  
Fuente: Autores

El control de “*Smart Grids*” se realiza en tres fases: software, hardware y dispositivos físicos. El desarrollo del algoritmo *VNS-DEEPSO* permite realizar mejoras en la fase de software. Debido a que en el proceso de la simulación del hardware pueden presentarse algunos problemas, tales como tiempos de retardo, pérdida de paquetes, y fallas electrónicas o mecánicas en dispositivos físicos; a medida que continúe el desarrollo de las investigaciones en esta área, se podrán realizar simulaciones que incluyan fallas en la fase de hardware y dispositivos físicos [15].

## V. CONCLUSIONES

El problema de operación óptima de una “*Smart Grids*” fue resuelto utilizando el algoritmo *VNS-DEEPSO* con el mejor ranking índice de la competencia de 18,21 este valor de comparación se calculó sumando el promedio de 20 pruebas más su desviación estándar. Los siguientes algoritmos denominados *EVDEEPSO*, *CESO* y *PSO-GBE*, presentaron una diferencia del 7% y del 37% con respecto al algoritmo ganador, respectivamente. Acorde a los resultados, el algoritmo *VNS-DEEPSO* demostró ser el más eficiente en minimizar los costos operacionales y maximizar los ingresos de una “*Smart Grids*” (Tabla 3).

Conforme con los resultados obtenidos en la competencia por el algoritmo *VNS-DEEPSO* se resaltan dos casos: 1) La generación de potencia distribuida que se mantiene fluctuando alrededor de los 20 kW, con dos picos, el primero a la hora 13, bajando levemente hasta el valle en la hora 15, después llega a su segundo pico (punto máximo) de 103,8 kW a la hora 17 y cae levemente hasta un valor un poco sobre 20 kW. 2) La carga de 24 vehículos eléctricos que mantiene en su mayor carga únicamente entre la hora 2 y la hora 5 fluctuando alrededor de 28 kW.

El algoritmo *VNS* se mejoró con el método de coordenada cíclica y el método de Fibonacci, este se unió con el algoritmo *DEEPSO*. La unión entre estos dos algoritmos se realizó en un valor de 48,8% de *VNS* y 51,2% de *DEEPSO*, lo cual mejoró las funciones del agregador para administrar las redes inteligentes.

## REFERENCIAS

- [1] M. Tuballa and M. Abundo, “A review of the development of Smart Grid technologies,” *Renewable and Sustainable Energy Reviews*, vol. 59, no. 1, pp. 710–725, Jun. 2016. Doi: <https://doi.org/10.1016/j.rser.2016.01.011>
- [2] F. Lezama, J. Soares, Z. Bale, J. Rueda, S. Rivera and I. Elrich, “2017 IEEE competition on modern heuristic optimizers for smart grid operation: Testbeds and results,” *Swarm and Evolutionary Computation*, vol. 44, no. 1, pp. 420–427, Feb. 2019. Doi: <https://doi.org/10.1016/j.swevo.2018.05.005>
- [3] O. Dzobo, A. M. Shehata and C. L. Azimoh, “Optimal economic load dispatch in smart grids considering uncertainty”. In *AFRICON*, 2017 IEEE., Cape Town, South Africa, Sept. 18–20, 2019, pp. 1277–1282. Doi: <https://doi.org/10.1109/AFRCON.2017.8095666>
- [4] F. Lezama, J. Soares, Z. Bale and J. Rueda, “WCCI 2018 Competition Evolutionary Computation in Uncertain Environments: A Smart Grid Application. Test bed: Optimal scheduling of distributed energy resources considering uncertainty of renewables, EVs, load forecast and market prices,” in *Proc. IEEE WCCI*, Rio de Janeiro, Brazil, Jul. 8–13, 2018, pp. 1–19. Available in: <http://www.gecad.isep.ipp.pt/WCCI2018-SG-COMPE-TITION/>
- [5] Y. Jin and J. Branke, “Evolutionary Optimization in Uncertain Environments—A Survey,” *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005. Doi: <https://doi.org/10.1109/TEVC.2005.846356>
- [6] M. Gendreau and J.-Y. Potvin, “Handbook of Metaheuristics,” vol. 2, Switzerland: Springer Science+Business Media, 2010. Doi: <https://doi.org/10.1007/978-1-4419-1665-5>
- [7] Y. Zhao, J. Cao, X. Lai, C. Yin and T. Chen, “Ensemble based constrained-optimization extreme learning machine for landmark recognition,” *2015 34th Chinese Control Conference (CCC)*, Hangzhou, China, 28–30 Jul. 2015, pp. 3884–3889. Doi: <https://doi.org/10.1109/ChiCC.2015.7260239>
- [8] S. Hui and P. N. Suganthan, “Ensemble and Arithmetic Recombination-Based Speciation Differential Evolution for Multimodal Optimization,” in *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 64–74, Jan. 2016. Doi: <https://doi.org/10.1109/TCYB.2015.2394466>
- [9] X. Li, S. Ma and Y. Wang, “Multi-Population Based Ensemble Mutation Method for Single Objective Bilevel Optimization Problem,” in *IEEE Access*, vol. 4, no. 1, pp. 7262–7274, Oct. 2016. Doi: <https://doi.org/10.1109/ACCESS.2016.2617738>
- [10] V. Miranda and R. Alves, “Differential Evolutionary Particle Swarm Optimization (DEEPSO): A Successful Hybrid,” *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, Ipojuca, Brazil, 8–11 Sept. 2013, pp. 368–374. Doi: <https://doi.org/10.1109/BRICS-CCI-CBIC.2013.68>
- [11] D. G. Mayer, B. P. Kinghorn and A. A. Archer, “Differential evolution – an easy and efficient evolutionary algorithm for model optimisation,” *Agricultural Systems*, vol. 83, no. 3, pp. 315–328, Mar. 2005. Doi: <https://doi.org/10.1016/j.agsy.2004.05.002>
- [12] F. Marini and B. Walczak, “Particle swarm optimization (PSO). A tutorial,” *Chemometrics and Intelligent Laboratory Systems*, vol. 149, Part B, pp. 153–165, Dec. 2015. Doi: <https://doi.org/10.1016/j.chemolab.2015.08.020>
- [13] P. Hansen and N. Mladenović, “Variable neighborhood search: Principles and applications,” *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, May. 2001. Doi: [https://doi.org/10.1016/S0377-2217\(00\)00100-4](https://doi.org/10.1016/S0377-2217(00)00100-4)
- [14] E. Vargas, L. Macedo, P. Bueno and R. Romero, “A VNS algorithm for the design of supplementary damping controllers for small-signal stability analysis,” *International Journal of Electrical Power & Energy Systems*, vol. 94, no. 1, pp. 41–56, Jan. 2018. Doi: <https://doi.org/10.1016/j.ijepes.2017.06.017>
- [15] R. Brandl, P. Kotsampopoulos, G. Lauss, M. Maniatiopoulou, M. Nuschke, J. Montoya and D. Strauss-Mincu, “Advanced Testing Chain Supporting the Validation of Smart Grid Systems and Technologies,” in *2018 IEEE Workshop on Complexity in Engineering*, Florence, Italy, Oct. 10–12, 2018, pp. 1–6. Doi: <https://doi.org/10.1109/CompEng.2018.8536223>

**Pedro Julián García** MSc. en Ingeniería Electromecánica de la Universidad Pedagógica y Tecnológica de Colombia (2012); Especialista en Automatización Industrial de la Universidad Francisco de Paula Santander de Ocaña (2018); MSc. en Ingeniería Mecánica en la Universidad Nacional de Colombia (2016); PhD (c). en Ingeniería Eléctrica en la Universidad Nacional de Colombia; profesor ocasional de la UNAL y la UFPSO, ingeniero gestor de eficiencia energética y gestor Tecnoparque SENA nodo Ocaña. **ORCID:** <http://orcid.org/0000-0002-8042-1299>

**Julián Cantor** es Ingeniero Electricista de la Universidad Nacional de Colombia, sede Bogotá. Actualmente realiza la evaluación técnica y financiera de proyectos de expansión de red eléctrica y de gas en la Unidad de Planeación Minero-Energética – UPME.

**Camilo Cortés** Ph.D. Ingeniero Electricista de la Universidad Nacional de Colombia (2000) y Doctor en Ingeniería Eléctrica de la Universidad Nacional de San Juan, Argentina (2005), con beca del Servicio Alemán de Intercambio Académico DAAD. Estudiante doctoral visitante de la Universidad de Ciencias Aplicadas de Giessen-Friedberg y el NLÖ de Alemania (2002). Visitante posdoctoral de la Universidad Católica de Lovaina KUL, Bélgica (2006). Profesor de la Universidad de la Salle de 2005 a 2007. Posdoctorado en el Illinois Institute of Technology, USA (2015-2016). Profesor Asociado de la Universidad Nacional de Colombia desde el año 2008, y Embajador Científico del DAAD desde el año 2017. **ORCID:** <http://orcid.org/0000-0002-0986-3975>

**María Alejandra Guzmán** es ingeniera mecánica de la Universidad Nacional de Colombia. Posteriormente obtuvo su título de maestría en Automatización Industrial en la Universidad Nacional de Colombia y el título de Doctora en Ingeniería Mecánica en la Universidad de Sao Paulo, Brasil. Es profesora del Departamento de Ingeniería Mecánica y Mecatrónica de la Universidad Nacional de Colombia desde hace 21 años. Su área de interés es la optimización mono y multiobjetivo bio-inspirada.

**Sergio Rivera Rodríguez** Ph.D. Ingeniero Electricista de Universidad Nacional de Colombia (2001); Especialista en Ingeniería Eléctrica con Énfasis en Sistemas de Distribución (2004), PhD en Ingeniería Eléctrica del Instituto de Energía Eléctrica, Universidad Nacional de San Juan (2011). Postdoctorado Asociado en el MIT – Massachusetts Institute of Technology (2013); Postdoctoral Fellow en el MIST – Masdar Institute of Science and Technology (2014). Profesor en Universidad Nacional en el área de sistemas de potencia y máquinas eléctricas (2014). **ORCID:** <http://orcid.org/0000-0002-2995-1147>