

Ajuste posinomial combinando Enjambre de Partículas y Métodos de Programación No-lineal

Posynomial fitting combining Particle Swarm and Nonlinear Programming Methods

Nabote António-Magaia^I, Isidro Luis Alemán-Romero^{II}, José Arzola-Ruiz^{II}
Osmel Martínez-Valdés^{III}

I. Instituto Superior Politécnico de Songo. Tete. Mozambique

II. Universidad Tecnológica de La Habana. Centro de Estudios de Matemática. La Habana. Cuba

III. Empresa Acinox Ingeniería S.A. La Habana. Cuba

Correo electrónico: ialeman@cemat.cujae.edu.cu

Este documento posee una [licencia Creative Commons Reconocimiento-No Comercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/)



Recibido: 3 de enero de 2017

Aceptado: 15 de abril de 2017

Resumen

El desempeño de los métodos determinísticos de programación no-lineal es altamente sensible de la selección de la aproximación inicial, en particular cuando se trata de la obtención de un óptimo global, como es requerido en el ajuste posinomial. El presente trabajo tuvo como objetivo, generar la aproximación inicial a través de la metaheurística enjambre de partículas, para mejorar el desempeño de los métodos de programación no-lineal en el ajuste posinomial. A partir de pruebas no paramétricas que se realizaron para la comparación de los resultados obtenidos en los problemas considerados, se observó que las diferencias existentes entre los

resultados del ajuste posinomial, con y sin la aplicación de la propuesta de mejora, son significativas. Además, se constató que la metodología de mejora desarrollada permite obtener un buen desempeño de los métodos de programación no-lineal en el ajuste posinomial, lo que redundó en la obtención de modelos posinomiales de calidad.

Palabras claves: ajuste posinomial, programación no-lineal, optimización por enjambre de partículas, programación geométrica.

Abstract

The performance of deterministic methods of nonlinear programming is highly sensitive to the selection of the initial approximation, particularly when it comes to obtaining an overall optimum, as required in the posynomial fitting. The present work aimed to generate the initial approximation through of particle swarm metaheuristic, to improve the performance of non-linear programming methods in posynomial fitting. From the non-parametric tests performed for the comparison of the results obtained in the considered problems, it was observed that the

differences between the results of the posynomial fitting, with and without the application of the improvement proposal, are significant. In addition, it was found that the improvement methodology developed allows to obtain a good performance of the non-linear programming methods in the posynomial fitting, which results in the obtaining of quality posynomial models.

Key words: posynomial fitting, nonlinear programming, particle swam optimization, geometric programming.

Introducción

El problema de ajuste posinomial constituye una tarea de optimización no-lineal y no convexa y, por lo tanto, no se garantiza la obtención del óptimo global [1]. La calidad de ajuste posinomial a través de los métodos clásicos (determinísticos) de programación no-lineal (PNL), depende de la aproximación inicial y, desafortunadamente, no existe una forma clara para determinarla. Por otra parte, aunque se considere un espacio amplio de búsqueda, la metaheurística enjambre de partículas permite identificar soluciones próximas al óptimo global, por lo que su utilización en la generación de la aproximación inicial para los métodos de PNL, posibilita una rápida convergencia a soluciones aceptables. Por tanto, la combinación de estos dos métodos mejora el desempeño de los métodos de PNL y, consecuentemente, la calidad de ajuste posinomial. Los posinomiales obtenidos se pueden utilizar con carácter predictivo y en la formulación de problemas de la Programación Geométrica (PG), que tienen muchas aplicaciones en la ingeniería [2, 3].

El ajuste posinomial mediante el empleo de métodos de PNL, se desarrolla en [1], donde se minimiza la suma de cuadrados de los errores relativos. Entre los métodos de PNL más utilizados se encuentran: Gauss-Newton, Quasi-Newton y programación cuadrática secuencial (PCS). En el enfoque propuesto en [1], se propone solucionar el problema de ajuste posinomial mediante la generación aleatoria de varias aproximaciones iniciales y, posteriormente, se selecciona el posinomial que brinde la mejor aproximación [1, 4]. Para determinar las aproximaciones iniciales, primero se realiza un ajuste monomial preliminar, y luego se perturban sus exponentes y su coeficiente de forma aleatoria y, posteriormente, se considera la combinación

lineal de los monomiales generados. Además de la cantidad de tiempo que esta estrategia requiere, no se garantiza el logro del óptimo global [4].

Otro enfoque local de ajuste posinomial se basa en la combinación de los algoritmos genéticos (AG) y la programación cuadrática (PC) [5], pero a expensa de un alto costo computacional. Se hace necesario utilizar técnicas modernas que disminuyan el tiempo computacional y que logren buenos resultados, por lo que se decide utilizar la metaheurística enjambre de partículas, que ha brindado buenos resultados en numerosas aplicaciones.

Debido a complejidad de los problemas de ajuste posinomial, es frecuente la utilización de conjuntos de funciones monomiales, o bien, funciones max-monomiales para ajustar funciones convexas [6]. Para mejorar el error de ajuste por funciones max-monomiales, se utilizan en [7] técnicas estadísticas.

Del análisis anterior se constata que los enfoques desarrollados para el ajuste posinomial son locales y se basan en los métodos clásicos de PNL y en las técnicas heurísticas. La calidad de ajuste posinomial a través de los métodos clásicos de PNL, depende de la elección de la aproximación inicial para comenzar el proceso de búsqueda de los parámetros. Sin embargo, en la literatura estudiada no existe una forma clara para determinar una prometedora aproximación inicial para los métodos de PNL.

El objetivo de este artículo es generar una aproximación inicial a través del algoritmo de Optimización por Enjambre de Partículas (OEP) para mejorar el desempeño de los métodos clásicos de PNL en el ajuste posinomial.

Para lograr este propósito, se supone que si se utiliza el algoritmo de OEP para generar el punto de partida para iniciar los métodos de PNL, se podrá mejorar el desempeño de estos métodos y, consecuentemente, la calidad de ajuste posinomial.

Para la generación de la aproximación inicial, se implementa computacionalmente en Matlab, el algoritmo de OEP. Con respecto a los métodos de PNL, se usan las funciones `fmincon`, `fminunc` y `fminsearch` del Matlab, con condiciones impuestas para implementar los métodos de PCS, Quasi-Newton y Simplex, respectivamente.

Se aplican los métodos estadísticos para comparar los resultados de ajuste posinomial, con y sin la aplicación del algoritmo de OEP. Además, se realiza el diseño factorial 3^4 para obtener los datos que, posteriormente, se ajustan por modelos posinomiales, en el proceso de descripción de los comportamientos de la Eficiencia energética (Ef), Caudal másico (CM) y Temperatura de la zona de pirolisis (TZ), en una instalación experimental de gasificación tipo *downdraft*.

Se consideran dos ejemplos prácticos, donde se realiza el ajuste posinomial, con y sin la aplicación de la metaheurística. El primer ejemplo se corresponde con el problema de determinación de un modelo posinomial que sustituya al procedimiento de cálculo de la temperatura externa de una pared refractaria. El segundo es de determinación de modelos posinomiales que describan los comportamientos de Ef , CM y TZ . Se realiza la comparación directa de los mejores resultados obtenidos y se observa que la combinación propuesta en el trabajo brinda los mejores resultados de ajuste posinomial. Además, se evidencia, a través de las pruebas no paramétricas de Mann-Whitney, que las diferencias entre los errores obtenidos, son significativas.

En el trabajo, se constata que la generación de la aproximación inicial a través del algoritmo de OEP mejora el desempeño de los métodos de PNL en el ajuste posinomial

Métodos

Considerando que no existe una forma clara para determinar la aproximación inicial en la aplicación de los métodos de PNL en el ajuste posinomial, en este trabajo se pretende analizar el desempeño de estos métodos cuando se considere el algoritmo de OEP para generar la aproximación inicial. Para la realización de esta tarea, se requiere la selección del criterio de proximidad entre las funciones aproximada y aproximante, la implementación del algoritmo de OEP, la especificación de los métodos de PNL a considerar y, por el último, la técnica para la validación de los resultados obtenidos.

La investigación se realiza mediante la combinación de algunos métodos de la investigación teórica y el diseño factorial 3^4 . Se emplea el método de Análisis - Síntesis consistente en el estudio pormenorizado y la descomposición de las teorías sobre el ajuste posinomial. El método Hipotético – Deductivo se emplea para la formulación y la demostración de la hipótesis de investigación a partir de la comparación de los resultados obtenidos. Para la comparación de los resultados se utilizan las pruebas no paramétricas pues, a priori los resultados no siguen ningún tipo de distribución de probabilidad. Se plantean las siguientes hipótesis H_0 y H_1 :

$$H_0: \mu_1 = \mu_0$$

No existen diferencias significativas entre los errores de ajuste posinomial que se obtienen a través de métodos de PNL, con y sin el empleo del algoritmo de OEP para generar la solución inicial.

$$H_1: \mu_1 \neq \mu_0$$

Existen diferencias significativas entre los errores de ajuste posinomial que se obtienen a través de métodos de PNL con y sin el empleo del algoritmo de OEP para generar la solución inicial.

La regla de decisión consiste en rechazar la hipótesis nula si el valor de la probabilidad asociada es menor que el nivel de significación de la prueba $\alpha = 0,05$

Con el objetivo de obtener los datos que describan los comportamientos de las variables *Ef*, *CM* y *TZ*, se realizó un diseño factorial 3^4 . Se consideró el análisis de estos tres rendimientos independientes y se tomaron 3 réplicas y 4 factores, denominados: Poder calórico de la biomasa (*PC*), Humedad de la biomasa (*HB*), Apertura de la válvula de aire (*AV*) y la Cantidad de biomasa (*CB*). Para las variables independientes *PC*, *HB*, *AV* y *CB* se asumieron los valores de los siguientes conjuntos {4000, 13800, 19000}, {12, 16, 20}, {25, 50, 100} y {1, 3, 5}, respectivamente.

El control local, se garantizó por el medio del control visual, donde se empleó un observador experimentado para verificar el cumplimiento de las operaciones previstas, según lo establecido a través de un *checklist*.

Ahora bien, para la realización de la tarea de aproximación de funciones, se debe especificar la estructura matemática de la clase de funciones candidatas. En este trabajo, las funciones de aproximación elegidas son los monomiales y posinomiales. Los monomiales se corresponden con la siguiente estructura matemática:

$$y = c x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} = c \prod_{i=1}^n x_i^{\alpha_i}, \quad c, x_i > 0, \alpha_i \in R, i = 1, 2, \dots, n. \quad (1)$$

Los posinomiales son modelos matemáticos de la forma:

$$g(x) = \sum_{k=1}^m c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \dots x_n^{\alpha_{nk}} = \sum_{k=1}^m c_k \prod_{i=1}^n x_i^{\alpha_{ik}}, \quad c_k, x_i > 0, \alpha_{ik} \in R, i = 1, 2, \dots, n; k = 1, 2, \dots, m. \quad (2)$$

Existen varias normas para establecer la proximidad entre las funciones original f y la aproximante g . En el presente trabajo, la proximidad entre f y g se instituye a través de la siguiente función:

$$\|f - g\| = \left(\frac{1}{N} \cdot \sum_{i=1}^N (f(x_i) - g(x_i))^2 \right)^{1/2} \quad (3)$$

Los métodos de PNL que se consideran para el ajuste posinomial son los implementados en las funciones *fmincon*, *fminunc* y *fminsearch* del Matlab. Para el análisis de los resultados, se consideran dos problemas: el primero es de determinación de un modelo posinomial con dos variables, que sustituya al procedimiento de cálculo de la temperatura externa de una pared refractaria; el segundo es de determinación de modelos posinomiales que describan el comportamiento de tres salidas independientes (*Ef*, *CM*, *TZ*), donde cada una de ellas depende de cuatro entradas (*PC*, *HB*, *AV*, *CB*), en una instalación experimental de gasificación de biomasa tipo *downdraft*.

Para la validación de los resultados, se comparan los resultados de ajuste posinomial, con y sin la aplicación de la metaheurística, de los datos correspondientes a los dos problemas anteriormente descritos.

La implementación computacional del algoritmo de OEP, se realizó en Matlab. A continuación se explican y se presentan las componentes de la variante del algoritmo de OEP, que se consideró en este trabajo.

Optimización por enjambre de partículas

El algoritmo de OEP es una metaheurística que frecuentemente se utiliza en la optimización global. Fue desarrollado por Kennedy y Eberhart (1995) y está inspirado en el movimiento colectivo de bandadas de pájaros, cardúmenes de peces o enjambre de abejas. Se desarrollaron varias variantes de OEP para asegurar la convergencia, la estabilidad y el desempeño de esta metaheurística. La convergencia y la estabilidad de OEP estándar dependen de factores tales como la dimensión del problema y la variante de OEP que se considere [8]. Otros factores importantes que influyen en la convergencia del algoritmo de OEP y en la calidad de las soluciones que este método brinda, son la inercia y el tipo de función de distribución de probabilidad que se utilice para la generación aleatoria de los vectores diagonales que multiplican las componentes social y cognitiva, en la ecuación de velocidad de las partículas [9]. Por tanto, se desarrollaron muchos estudios para la mutación de los operadores en la ecuación de actualización de la velocidad de las partículas. Otros estudios indican que para ciertos problemas, la variante de OEP que utiliza los pesos exponenciales para la inercia, ofrece mejores resultados que el uso de pesos lineales [10]. Por otra parte, la estrategia que se use para lograr una mayor diversidad de la población en el algoritmo de OEP, también tiene un papel importante en calidad de las soluciones [11].

En este trabajo se considera la variante del algoritmo de OEP modificado y que utiliza los pesos de la inercia como una función lineal y decreciente [12]. La ecuación que describe la actualización de la velocidad de las partículas es dada por:

$$V_i^{n+1} = w V_i^n + c_1 \text{rand}_1(0,1) \cdot (pbest_i - X_i^n) + c_2 \text{rand}_2(0,1) \cdot (gbest - X_i^n) \quad (4)$$

Donde, w es el peso de inercia, c_1 y c_2 son coeficientes positivos de ponderación. El parámetro w sirve de balance entre la capacidad de búsqueda global y local. Mientras mayor sea el valor de w se facilita la

búsqueda global, y en caso contrario la búsqueda local. Los coeficientes c_1 y c_2 se introducen para asegurar la convergencia del algoritmo de OEP. Los términos $rand_1(0,1)$ y $rand_2(0,1)$ son funciones para generar números aleatorios entre cero y uno. La partícula i y su velocidad en la iteración n , se representan por $X_i^n = [x_{i1}, x_{i2}, \dots, x_{iD}]$ y $V_i^n = [v_{i1}, v_{i2}, \dots, v_{iD}]$, respectivamente. La representación $pbest_i$ se corresponde con la anterior mejor ubicación de las partículas, mientras que $gbest$ es la posición de la mejor partícula, es decir, la partícula que ofrece el menor valor de la función objetivo.

La relación (4) está compuesta por tres términos: el primero (wV_i^n) es referente a la inercia; el segundo ($pbest_i - X_i^n$) es la parte cognitiva, que representa el autoaprendizaje a partir de la propia experiencia del vuelo; el tercero y último ($gbest_i - X_i^n$), es la componente social, que representa la colaboración en el aprendizaje de las partículas en las experiencias de vuelos colectivos. La dinámica de las trayectorias del vuelo de las partículas se presenta por la relación (5):

$$X_i^{n+1} = X_i^n + V_i^{n+1} \tag{5}$$

La función que se corresponde con la mutación del factor peso de inercia es dada por la siguiente relación:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{N} \cdot n \tag{6}$$

Donde, w_{\min} y w_{\max} son los valores mínimo y máximo del peso de inercia. Los parámetros n y N , representan la iteración y el número total de iteraciones prefijadas, respectivamente.

Para la generación de aproximaciones iniciales a través del algoritmo de OEP, en este trabajo se considerarán 600 iteraciones, 30 corridas y los factores de peso se corresponden con los siguientes valores: $w_1 = 0,001$, $w_2 = 0,03$ y $c_1 = c_2 = 2$.

Ahora se presenta el problema de ajuste posinomial y, posteriormente, se formula como una tarea de optimización no convexa.

Ajuste posinomial

Se supone que se conocen los puntos $(x^{(i)}, f^{(i)})$, $i = 1, 2, \dots, N$, donde $x^{(i)} \in R^n$ son vectores positivos y $f^{(i)}$ son constantes positivas. El problema de ajuste posinomial consiste en determinar los valores de los coeficientes $c_k > 0$ y de los exponentes $\alpha_{1k}, \alpha_{2k}, \dots, \alpha_{nk}$, tales que,

$$\sum_{k=1}^m c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \dots x_n^{\alpha_{nk}} = \sum_{k=1}^m c_k \prod_{i=1}^n x_i^{\alpha_{ik}} \approx f \tag{7}$$

El problema (7) se puede abordar de varias formas [13]. Una de las más empleadas se basa en transformaciones logarítmicas de las funciones involucradas de modo que se minimice la norma de la diferencia de dos funciones logarítmicas, es decir, se optimiza el siguiente problema:

$$\text{Minimizar} \left\| \log f - \log \sum_{k=1}^m e^{a_k y + b_k} \right\| \tag{8}$$

Donde, $a_k = [\alpha_{1k}, \alpha_{2k}, \dots, \alpha_{nk}]$, $y = [\log x_1, \log x_2, \dots, \log x_n]^T$, y $b_k = \log c_k$

Ahora bien, a partir de las propiedades de los posinomiales, el logaritmo de la suma de los exponenciales es una función convexa [13]. Sin embargo, en general el problema (8) no es convexo [2] y los métodos determinísticos que frecuentemente se utilizan para solucionarlo son locales y requieren de una buena aproximación inicial para que brinden soluciones aceptables [2]. En el enfoque de determinación de modelos posinomiales a partir de la perturbación de los exponentes del primer monomial generado, no se puede lograría un vector de exponentes nulos, constituyendo una restricción que afecta la estructura física del modelo posinomial y, por lo tanto, se puede perder informaciones en el proceso de ajuste de datos. Para evitar este problema, en este trabajo se propone añadir un parámetro no negativo en el modelo posinomial estándar, de tal forma que se optimice el siguiente problema:

$$\text{Minimizar} \left\| c_0 + \sum_{k=1}^m c_k \prod_{i=1}^n x_i^{\alpha_{ik}} - f \right\| \tag{9}$$

Para solucionar el problema (9) a partir de los métodos clásicos de PNL, se toman como puntos iniciales los vectores que se obtienen a partir de la perturbación de los parámetros del primero monomial generado. Es decir, se construye un posinomial inicial a partir de la combinación lineal de los monomiales previamente

generados y, todos ellos, próximos al primero obtenido [1]. El primer monomial se puede lograr fácilmente a través del método de mínimos cuadrados, tras la realización de transformaciones logarítmicas de las funciones involucradas.

Para la discusión y comparación de los resultados de ajuste posinomial a través de métodos de PNL, con y sin la utilización del algoritmo de OEP para generar la aproximación inicial, se consideran los dos ejemplos referidos con anterioridad y que se describen a continuación:

Ejemplo 1. Se considera el problema de la determinación del modelo posinomial que se aproxime al procedimiento de cálculo de la temperatura externa de una pared refractaria de un horno. La temperatura externa (*Text*) de la pared se considera como una función que depende del espesor de su primera capa (*S*) y de la temperatura interna (*Tint*). Para la generación de la base de datos, se considera la información de la tabla 1:

Tabla 1, Perfil de la pared refractaria

Material de la primera capa (Chamota)	Valor numérico de los materiales	Material de la segunda capa (Dinas)	Valor numérico de los materiales
Coefficiente de termoconductividad	0,7+0,00640 T	Coefficiente de termoconductividad	0,815+0,000676 T
Espesor mínimo de la capa	100 mm	Espesor de la capa	246 mm
-	-	Máxima temperatura interna	1300 °C
-	-	Mínima temperatura interna	1000 °C
-	-	Temperatura ambiental	30 °C

T representa la temperatura (en °C) de un perfil de la pared.

A partir del procedimiento de cálculo de la distribución de la temperatura en la pared, se genera un fichero de datos compuesto por 360 puntos, que se logran al fijar las variaciones $\Delta T_{int} = 20^\circ\text{C}$ y $\Delta S = 6\text{ mm}$, donde *Tint* y *S* varían de 1020 °C a 1300 °C y 108 mm a 246 mm, respectivamente. Una vez que se logra la base de datos, la tarea consiste en buscar un modelo posinomial para estimar *Text* como una función que depende de *S* y *Tint*.

Ahora bien, para la descripción del comportamiento de la temperatura externa a partir de modelos posinomiales, se seleccionó un posinomial con tres términos. A continuación se presenta la tabla de los resultados del ajuste posinomial a través de los métodos de PNL, con y sin la aplicación de la metaheurística:

Tabla 2. Comparación de los errores brindados por los métodos clásicos de PNL implementados en Matlab, con y sin la aplicación de la metaheurística.

Elementos de comparación	Generación aleatoria de 30 aproximaciones iniciales			Aplicación del algoritmo de OEP para generar 30 aproximaciones iniciales		
	fmincon	fminunc	fminsearch	fmincon	fminunc	fminsearch
Mejor	1,3746	1,3803	1,3802	1,3700	1,3697	1,3699
Peor	21,5187	17,8833	22,3256	1,3751	1,3721	1,3721
Media	2,1477	2,0319	2,1364	1,3722	1,3709	1,3709
DE	3,6610	3,0430	3,8255	4,41E-3	6,52E-4	6,56E-4
CV	1,7046	1,4976	1,7906	3,66E-3	4,76E-4	4,79E-4
IC de 95%	2,1477 +/- 1,3669	2,0319 +/- 1,1361	2,1364 +/- 1,4283	1,3722 +/- 1,53E-3	1,3709 +/- 3,0191E-3	1,3709 +/- 2,45E-4

Las abreviaciones DE y CV se corresponden con la desviación estándar y el coeficiente de variación, respectivamente.

El mejor modelo posinomial que se obtuvo ofrece un error de 1,3697, explica 99,78 % de la variabilidad de la temperatura externa y se corresponde con la siguiente expresión analítica:

$$Text(S, T_{int}) = 2,157147 \cdot 10^{-5} + 0,30669916 \cdot S^{-0,31733679} \cdot T_{int}^{1,1790974} + 1879,8173 \cdot S^{-0,19018784} \cdot T_{int}^{1,1412907}$$

Ejemplo 2. Ahora se considera el problema de determinación de un modelo matemático que describa los indicadores de desempeño de una instalación experimental de gasificación tipo downdraft. Con este fin, se realiza un experimento 3^4 , con 3 réplicas y, por lo tanto, se realizan 243 experiencias. Así, se buscan modelos posinomiales para ajustar los comportamientos de las siguientes salidas independientes: Eficiencia energética (*Ef*), Caudal másico (*CM*) y Temperatura de la zona de pirólisis (*TZ*). Las variables independientes del proceso son: Poder calórico de la biomasa (*PC*), Humedad de la biomasa (*HB*), Apertura de la válvula de aire (*AV*) y la Cantidad de biomasa (*CB*).

Para la descripción del comportamiento de la eficiencia se considera un posinomial con 4 términos, mientras que para *CA* y *TZ*, se seleccionan los modelos posinomiales con tres términos. A continuación se presentan los resultados del ajuste posinomial a través de los métodos clásicos, con y sin la aplicación del algoritmo de OEP para generar las aproximaciones iniciales, para las tres variables de salidas:

Tabla 3. Comparación de los errores brindados por los métodos determinísticos implementados en Matlab, con y sin la aplicación de la metaheurística, en la estimación del *CM*

Variable de salida	Elementos de comparación	Generación aleatoria de 30 aproximaciones iniciales			Aplicación del algoritmo de OEP para generar 30 aproximaciones iniciales		
		fmincon	fminunc	fminsearch	fmincon	fminunc	fminsearch
CM	Mejor	0,0586	0,0556	0,0568	0,0458	0,0457	0,0466
	Peor	0,0856	0,8256	1,8419	0,0666	0,0610	0,1026
	Media	0,3271	0,3107	0,3454	0,0512	0,0512	0,0562
	DE	0,2028	0,1948	0,3406	0,0052	0,0043	0,0115
	CV	0,6198	0,6270	0,9862	0,1020	0,0845	0,2042
	IC de 95%	0,0856 +/- 0,0770	0,3107 +/- 0,0740	0,3454 +/- 0,1293	0,0512 +/- 0,0020	0,0512 +/- 0,0016	0,0562 +/- 0,0044

El mejor modelo que se logró ofrece un error de 0.0457, con el coeficiente de determinación múltiple $R^2 = 99.32\%$ y se corresponde con la siguiente expresión matemática:

$$CM(p, h, a, c) = 1.043877 \cdot 10^{-6} + 1.1889231 \cdot p^{-0.4420433} \cdot h^{-8.4989607E-17} \cdot a^{0.89985633} \cdot c^{3.2297288E-18} + 9.495935 \cdot p^{-1.264803} \cdot h^{0.11862749} \cdot a^{1.4100255} \cdot c^{0.002215259}$$

Donde, *p*, *h*, *a* y *c* representan *PC*, *HB*, *AV* y *CB*, respectivamente.

A continuación se presenta la tabla de comparación de los errores brindados por los métodos determinísticos, con y sin la aplicación de la propuesta de mejora, en la estimación de *Ef*.

Tabla 4. Comparación de los errores brindados por los métodos determinísticos implementados en Matlab, con y sin la aplicación de la propuesta de mejora, en la estimación de *Ef*

Variable de salida	Elementos de comparación	Generación aleatoria de 30 soluciones iniciales			Aplicación del algoritmo de OEP para generar 30 soluciones iniciales		
		fmincon	fminunc	fminsearch	fmincon	fminunc	fminsearch
Ef	Mejor	3,6918	3,6925	3,6921	2,749	2,7447	3,7448
	Peor	4,0715	4,1164	4,1296	4,6307	3,6552	4,5169
	Media	3,7799	3,7955	3,7942	3,7035	3,5364	3,5840
	DE	0,0827	0,090	0,0912	0,3014	0,0518	0,1933
	CV	0,0219	0,0237	0,0240	0,0814	0,0146	0,0539
	IC de 95%	3,7799 +/- 0,0309	3,7955 +/- 0,0335	3,7942 +/- 0,0340	3,7035 +/- 0,1125	3,5364 +/- 0,0193	3,5840 +/- 0,0722

El modelo posinomial que se logró brinda un error de 2.7447, explica 83.28% de variabilidad de la eficiencia y tiene la siguiente expresión analítica:

$$Ef(p, h, a, c) = 3.7813422 + 107.14686 p^{-0.041015326} \cdot h^{0.03876581} \cdot a^{-0.50735639} \cdot c^{0.15571229} + 1.0281399 \cdot 10^{-5} \cdot p^{1.4690685} \cdot h^{-0.13430008} \cdot a^{0.13838582} \cdot c^{-0.022974422} + 552.23722 p^{-0.64085979} \cdot h^{-0.74538772} \cdot a^{0.80481129} \cdot c^{0.056313396}$$

Se presenta la tabla de comparación de los errores brindados por los métodos determinísticos, con y sin la aplicación de la propuesta de mejora, en la estimación de *TZ*.

Tabla 5. Comparación de los errores brindados por los métodos determinísticos implementados en Matlab, con y sin la aplicación de la metaheurística, en la estimación de *TZ*

Variable de salida	Elementos de comparación	Generación aleatoria de 30 aproximaciones iniciales			Aplicación del algoritmo de OEP para generar 30 aproximaciones iniciales		
		fmincon	fminunc	fminsearch	fmincon	fminunc	fminsearch
TZ	Mejor	17,7892	14,7669	14,7658	14,1456	14,1346	14,1358
	Peor	22,8871	30,3652	36,0506	15,4390	14,7188	14,7188
	Media	16,2922	16,4301	16,6278	14,5030	14,3215	14,3230
	DE	1,9231	3,3518	4,2068	0,2791	0,1175	0,1172
	CV	0,1180	0,2040	0,2530	0,0192	0,0082	0,082
	IC de 95%	16,2922 +/- 0,7180	16,4301 +/- 1,2514	16,6278 +/- 1,5607	14,5030 +/- 0,1042	14,3215 +/- 0,0439	14,3230 +/- 0,0438

El modelo posinomial que se logró ofrece un error de 14.1346, $R^2 = 70.05$ y se corresponde con la siguiente expresión analítica:

$$TZ(p, h, a, c) = 7.6065828 + 92.516411 p^{0.11529784} \cdot h^{-0.15232088} \cdot a^{-0.010387912} \cdot c^{0.01942838} + 0.58606827 p^{0.56494931} \cdot h^{-0.67627429} \cdot a^{-0.60402147} \cdot c^{1.5113326}$$

Resultados y Discusión

Al aplicar las pruebas no paramétricas de Mann-Whitney, se observó que las diferencias en las tablas 2, 3, 4 y 5, entre los errores de ajuste posinomial obtenidos, con y sin la aplicación de la metaheurística, son significativas. A este resultado se arriba, empleando el software SPSS Statistics, al observar que el valor de la probabilidad asociada fue menor que el nivel de significación de la prueba $\alpha = 0,05$ en todos los ejemplos considerados. Lo anterior conlleva a que se debe rechazar la hipótesis nula en todos los problemas aquí tratados, o sea, hay evidencias estadísticas suficientes para rechazar la idea de que no existen diferencias significativas entre los errores de ajuste posinomial, obtenidos con y sin la aplicación de la metaheurística, con un nivel de confianza de 95 %. En las tablas 6 a la 9 se presentan los resultados de las pruebas no paramétricas de Mann – Whitney en el análisis de los errores que se generaron a través de las funciones *fminsearch* y *fmincon*, con y sin la aplicación de la metaheurística, en la estimación de *CM* y de *Ef*.

Tabla 6, Rangos medios de los errores en la estimación de *CM*

Ranks			
Rango	N	Mean Rank	Sum of Ranks
<i>fminsearch</i> 1	30	16,27	488,00
2	30	44,73	1342
Total	60		

Tabla 8, Rangos medios de los errores en la estimación de *Ef*

Ranks			
Rango	N	Mean Rank	Sum of Ranks
<i>fmincon</i> 1	30	21,97	659,00
2	30	39,03	1171,00
Total	60		

Tabla 7, Estadísticas de la prueba para los errores en la estimación de *CM*

Test Statistics ^a	
	<i>fminsearch</i>
Mann-Whitney U	23,000
Wilcoxon W	488,000
Z	-6,313
Asymp. Sig. (2-Tailed)	,000

a. Grouping Variable: Rango

Tabla 9, Estadísticas de la prueba para los errores en la estimación de *Ef*

Test Statistics	
	<i>fminsearch</i>
Mann-Whitney U	194,000
Wilcoxon W	659,000
Z	-3,785
Asymp. Sig. (2-Tailed)	,000

a. Grouping Variable: Rango

Se observa en las tablas 7 y 9 que el valor de la probabilidad asociada es menor que 0.05, lo que significa que se debe rechazar la hipótesis nula.

Ahora se ofrece una tabla de resumen de los mejores resultados de ajuste posinomial, que se lograron en cada uno de los problemas considerados:

Tabla 10. Comparación de los errores brindados por los métodos de PNL implementados en Matlab, con y sin la aplicación del algoritmo de OEP para generar la aproximación inicial

Ejemplo	Variables independientes	Mínimo error que se logró sin la aplicación del algoritmo de OEP			Mínimo error que se logró con la aplicación del algoritmo de OPE		
		<i>fmincon</i>	<i>fminunc</i>	<i>fminsearch</i>	<i>fmincon</i>	<i>fminunc</i>	<i>fminsearch</i>
1	<i>Text</i>	1,3746	1,3803	1,3802	1,3700	1,3697	1,3699
2	<i>CM</i>	0,0586	0,0556	0,0568	0,0458	0,0457	0,0466
	<i>Ef</i>	3,6918	3,6925	3,6921	2,7490	2,7447	2,7448
	<i>TZ</i>	17,7892	14,7669	14,7658	14,1456	14,1346	14,1358

A partir de la tabla 10, se observa que cuando se utiliza el algoritmo de OEP para determinar la aproximación inicial para los métodos de PNL, se logran menores errores de ajuste, que los que se obtienen mediante la generación aleatoria de aproximaciones iniciales. Este resultado fortalece la idea de que la generación de la aproximación inicial a través del algoritmo de OEP mejora el desempeño de los métodos de PNL en el ajuste posinomial.

En el trabajo de Xue [4] en el ejemplo de la pagina 59 se realizó una aproximación monomial y se logró un error máximo de 0,1180, mucho mayor que 0,0324, que se obtuvo con la aplicación de la metodología desarrollada en este trabajo.

En el trabajo de Hoberg y Abbeel [14], al aproximar la función del ejemplo 6.1 por una función max-monomial con 2 términos, se logró un error de 0,0052 mostrado en la tabla 1. Por otra parte, con el empleo de la metodología propuesta en este trabajo se logró un posinomial con dos términos que ofreció un error de 0.0039.

La metodología desarrollada permite aprovechar:

- La capacidad de exploración del algoritmo de OEP para lograr un acercamiento al óptimo global del ajuste posinomial, que se toma como aproximación inicial para los métodos de PNL, es decir, la metaheurística enjambre de partículas asume el papel de método predictor.

- La capacidad de acercamiento de los métodos de PNL para lograr el mejoramiento de la aproximación inicial dada, que redundará en la obtención de modelos de ajuste posinomial de calidad, es decir, los métodos de PNL asumen el papel de métodos correctores.

En caso de no obtenerse una mejoraría al aplicar los métodos de PNL, esto indicaría que la solución obtenida mediante el algoritmo de OEP puede ser de buena calidad. Un mejoramiento adicional se podría lograr variando el parámetro w (peso de inercia) del algoritmo de OEP, lo que conllevaría a emplear este algoritmo a su vez como un método corrector. Al mismo tiempo, todo lo anterior incentiva a investigar otras metaheurísticas en un papel de método corrector y/o predictor.

Los modelos posinomiales, desarrollados por intermedio de la metodología descrita, se pueden aplicar en la aproximación de funciones log-log-convexas y en el ajuste de datos empíricos o de simulaciones tipo caja negra. Además se pueden utilizar para la realización de predicciones y para modelar y solucionar los problemas de optimización no-lineal y no convexa mediante la aplicación de la PG.

Conclusiones

Presentado en el trabajo el algoritmo de OEP y se consideró como complemento de los métodos clásicos de PNL en el ajuste posinomial. El algoritmo de OEP logra identificar aproximaciones iniciales de muy buena calidad para iniciar los métodos de PNL, lográndose mejores resultados de ajuste posinomial, que los obtenidos tomando aproximaciones iniciales aleatorias. Resultó que la generación de la aproximación inicial a través del algoritmo de OEP mejora el desempeño de los métodos de PNL en el ajuste posinomial.

Referencias

- Boyd S, Kim S, Vandenberghe L, et al. A tutorial on Geometric Programming. *Optimization and Engineering*. 2007;8(1):67-127.
- Hobug W, Abbel P. Geometric programming for aircraft design optimization. *AIAA Journal*. 2014;52:1-13.
- Boyd S, Kim S, Patil D, et al. Digital Circuit Optimization via Geometric Programming. *Operations Research*. 2005;53(6):899-932.
- Xue X. *Electronic Systems Optimization Design via GP-Based Surrogate Modelling*. North Carolina, USA: North Carolina State University; 2012. [Citado: julio de 2016] Disponible en: <http://repository.lib.ncsu.edu/ir/bitstream/1840.16/7710/1/etd.pdf>
- Aggarwal V, O'Reilly U. Simulation - Based Reusable Posynomial Models for MOS Transistor Parameters In: *Design, Automation and Test in Europe Conference and Exhibition*; Nice, Francia: Date-Conference.com; 2007. [Citado julio de 2016] Disponible en: https://www.date-conference.com/proceedings-archive/PAPERS/2007/DATE07/PDFFILES/01.5_1.PDF
- Kim J. Convex piecewise-linear modeling method for circuit optimization via geometric programming. *IEEE Transactions Computer-Aided Design of Integrated Circuits and Systems*. 2010;29(11):1823-7.
- Hannah A, Dunson B. Multivariate convex regression with adaptive partitioning. *Journal of Machine Learning Research*. 2013;14:3261-94.
- Bonyadi M, Michalewicz Z. Analysis of stability, local convergence and transformation sensitivity of a variant of particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*. 2015;20(3):370-85.
- Bonyadi M, Li X. An analysis of the velocity updating rule of the particle swarm optimization algorithm. *Journal Heuristics*. 2014;20(4):417-52.
- Ting T, Shi Y, Cheng S, et al. Exponential inertia weight for particle swarm optimization. In: *Third International Conference on Swarm Intelligence*; China: Researchgate.net; 2012. [Citado diciembre de 2016] Disponible en: https://www.researchgate.net/publication/262389715_Exponential_Inertia_Weight_for_Particle_Swarm_Optimization
- Shi C, Shi Y, Qin Q. Population diversity of particle swarm optimizer solving single and multi-objective problems. *International Journal of Swarm Intelligence Research*. 2012;3(4):23-60.
- Shi Y, Eberhart R. A modified particle swarm optimizer. In: *IEEE International Congress on Computational Intelligence*; Anchorage, USA: IEEE; 1998. [Citado diciembre de 2016] Disponible en: <https://pdfs.semanticscholar.org/5061/72b0e0dd4269bdcfe96dda9ea9d8602bbfb6.pdf>
- Boyd S, Vandenberghe L. *Convex Optimization*. New York, USA: Cambridge University Press; 2004.
- Hobug W, Abbel P. Data fitting with geometric-programming-compatible softmax functions. *Optimization and Engineering*. 2016;14(4):897-918.