CIENCIA 13(2), 193 - 204, 2005
Maracaibo, Venezuela

Algoritmo de optimización global eficiente con modelo aditivo acoplado

Juan Colmenares Díaz y Salvador Pintos Mantegani
Instituto de Cálculo Aplicado. Facultad de Ingeniería. Universidad del Zulia. Maracaibo, Venezuela

Recibido: 11-05-04 Aceptado: 15-03-05

Resumen

Importantes industrias en el mundo emplean herramientas computacionales para optimizar sus procesos productivos y construir diseños altamente eficientes que involucran funciones objetivo multimodales de alto costo computacional. Para resolver este tipo de problema se ha propuesto un algoritmo de optimización global bayesiana denominado EGO (*Efficient Global Optimization*) que se caracteriza por una sólida fundamentación teórica y por requerir muy pocas evaluaciones de la función objetivo, pero que -al mismo tiempo- presenta debilidades importantes: 1) supone que la función objetivo es estacionaria, lo cual puede conducir a resultados erróneos ya que existen muchos problemas de optimización donde no es posible garantizar dicha condición, 2) su propia naturaleza lleva a trabajar con matrices de correlación mal condicionadas y 3) la estimación del modelo DACE en cada iteración influye negativamente en el tiempo de ejecución. Para superar las debilidades antes señaladas este trabajo propone un nuevo algoritmo inspirado en EGO; presenta tres variantes del algoritmo (XEGO, REGO y NEGO) y compara su desempeño frente al programa SPACE (una implementación de EGO desarrollada en C++).

Palabras clave: Algoritmo EGO; funciones de alto costo computacional; modelo DACE; optimización global eficiente.

Efficient global optimization algorithm with additive and coupled model

Abstract

Important industries in the world use computational tools to optimize their production processes and create highly optimized designs that involve multimodal and computationally expensive objective functions. To solve this kind of problem a bayesian global optimization algorithm, called EGO (Efficient Global Optimization), has been proposed. EGO is characterized by solid theoretic foundations and to require very few evaluations of the objective function; however, it presents important weaknesses: 1) EGO supposes that the objective function is stationary, which may lead to erroneous results since there are many optimization problems that cannot guarantee this condition, 2) its own nature makes it work with Ill-conditioned matrices, and 3) DACE model estimation in each iteration affects its execu-

* Autor para la correspondencia. TeleFax: 58-261-7598411. E-mail: juancol@ica.luz.ve

tion time. This work proposes a new algorithm, inspired by EGO, to overcome the aforementioned weaknesses, presents three variants of the algorithm (XEGO, REGO and NEGO), and compares the performance of these variants and the program SPACE (an EGO implementation developed in C++).

Key words: Computationally expensive functions; DACE model; efficient global optimization; EGO algorithm.

1. Introducción

En importantes industrias como la petrolera, automotriz, aeroespacial y semiconductores, debido a la alta complejidad de sus procesos de producción y a las presiones competitivas, ha crecido la necesidad de emplear herramientas computacionales para la optimización de procesos productivos y la creación de diseños sofisticados e innovadores altamente eficientes; tales herramientas facilitan la exploración de alternativas y reducen la necesidad de construir prototipos. Como estos problemas de optimización, con frecuencia, involucran funciones objetivo multimodales implementadas mediante modelos de simulación de alto costo computacional, se han realizado importantes esfuerzos de investigación para desarrollar algoritmos de optimización global que requieran pocas evaluaciones de la función objetivo.

EGO (*Efficient Global Optimization*) (1) es un algoritmo de optimización global bayesiana caracterizado por una sólida fundamentación teórica y por ser altamente eficiente en términos de la cantidad requerida de evaluaciones de la función objetivo. Emplea el modelo DACE (2) para el ajuste sucesivo de superficies de aproximación y utiliza una figura de mérito para guiar el proceso de búsqueda garantizando un balance entre la exploración global y local del espacio. DACE es un método bayesiano para identificar modelos sustitutos de funciones determinísticas que presupone un modelo estacionario cuyo fundamento estadístico posibilita el cálculo de un predictor eficiente y de su incertidumbre asociada.

Pese a sus fortalezas EGO presenta algunas debilidades importantes. La principal de ellas es suponer que la función objetivo no presenta tendencias en la región de interés, es decir, asume que es estacionaria. Puesto que existen muchos problemas de optimización donde no es posible garantizar esta condición EGO puede arrojar resultados erróneos. Otras debilidades de EGO son que su propia naturaleza lo lleva a trabajar con matrices de correlación mal condicionadas, y que la reestimación del modelo DACE en cada iteración influye negativamente sobre su tiempo de ejecución.

Para superar las debilidades antes mencionadas, en este trabajo se propone un algoritmo, inspirado en EGO, que incluye las siguientes mejoras:

1. Un modelo aditivo en la estimación y acoplado en la optimización, para el ajuste sucesivo de superficies de aproximación cuando la función objetivo es no estacionaria.
2. Una estrategia para el manejo del mal condicionamiento de la matriz de correlación.
3. Un esquema de ajuste ligero del modelo DACE que favorece la velocidad de ejecución del algoritmo sin desmejorar la calidad de los resultados.

Se desarrollan tres variantes que difieren en el modelo de tendencia utilizado: 1) XEGO, no incluye modelo de tendencia, 2) REGO, emplea modelos de regresión lineal, y 3) NEGO, utiliza redes neuronales tipo perceptrón multicapa. Para evaluar su eficiencia se comparan frente al programa SPACE (3, 4) que es una implementación del algoritmo EGO desarrollada en C++ por M. Schonlau.

La organización de este artículo es la siguiente: las secciones 2 y 3 describen el método DACE y el algoritmo EGO, respectivamente. La sección 4 presenta el algoritmo propuesto, la sección 5 discute los resultados del estudio comparativo de casos y finalmente, la sección 6 establece las conclusiones.

2. Modelo DACE

La investigación científica de muchos fenómenos se realiza mediante el uso de modelos computacionales determinísticos. En particular, el interés se centra en modelos determinísticos continuos, invariantes en el tiempo, con entradas escalares y una única salida escalar. Con frecuencia los modelos empleados son computacionalmente costosos por lo que es necesario construir un modelo sustituto (de menor costo), a partir de un conjunto de datos de entrada/salida, capaz de predecir con precisión adecuada la respuesta del modelo original.

Típicamente se emplean modelos de regresión lineal (5) para ajustar superficies de respuesta. Estos modelos aproximantes no predicen exactamente los valores funcionales de puntos ya conocidos (no interpolan la data observada). Ellos presuponen la existencia de errores aleatorios independientes característicos de experimentos físicos, pero ausentes en el caso de modelos computacionales determinísticos.

Por ser el modelo de regresión lineal y el modelo original modelos determinísticos en función de $\mathbf{x}$ (vector de variables de entrada), el error (ξ) entre ellos también es función de $\mathbf{x}$. Más aún, si existen dos puntos $\mathbf{x}^{(i)}$ y $\mathbf{x}^{(j)}$ cercanos, los errores $\xi(\mathbf{x}^{(i)})$ y $\xi(\mathbf{x}^{(j)})$ deben ser similares. En consecuencia, en lugar de asumir que los errores son independientes, es razonable suponer que los errores están "correlacionados" y que la correlación es alta para puntos cercanos y baja para puntos lejanos. Esta es la idea básica sobre la cual se fundamenta el modelo DACE (2) que es un modelo bayesiano, adoptado del área de geoestadística (6), que trata la respuesta de modelos computacionales determinísticos como un proceso estocástico.

Supóngase que se ha evaluado una función determinística de k variables en n puntos y se denota la i-ésima observación como $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)},..., x_k^{(i)})$ cuyo valor funcional es $y^{(i)}$ donde $i=1,2,...,n$. Sacks et al. (2) proponen el siguiente modelo sustituto:

$$y(\mathbf{x}^{(i)}) = \sum_h \beta_h f_h(\mathbf{x}^{(i)}) + \xi(\mathbf{x}^{(i)}) \quad (i = 1, \ldots, n) \quad [1]$$

Este modelo considera a la respuesta determinística $y(\mathbf{x})$ como la realización de una función aleatoria $Y(\mathbf{x}^{(i)})$, incluye un modelo de regresión lineal y asume que el error $\xi(\boldsymbol{x}^{(i)})$ es un proceso estocástico que tiene una distribución normal con media cero, varianza σ^2 y covarianza dada por:

$$\operatorname{cov}\left[\xi(\boldsymbol{x}^{(i)}), \xi(\boldsymbol{x}^{(j)})\right] = \sigma^2 \exp\left(\sum_{h=1}^{k} \theta_h \left|x_h^{(i)} - x_h^{(j)}\right|^{ph}\right)$$
$$(\theta_h \geq 0, p_h \in [1,2]) \quad [2]$$

La función de correlación de la Ecuación [2] está definida bajo el supuesto de que cualquier comportamiento no estacionario puede ser modelado mediante el componente de regresión de la Ecuación [1]. Pese a no ser genuinamente la correlación entre n puntos, tiene todas las propiedades intuitivas necesarias: la correlación es cercana a uno cuando la distancia entre $\mathbf{x}^{(i)}$ y $\mathbf{x}^{(j)}$ es pequeña, y cercana a cero cuando la distancia entre los puntos es grande. El parámetro θ_h se interpreta como una medida de la variabilidad de la superficie en la dirección de la variable x_h. Nótese que si θ_h es muy grande entonces valores pequeños de $|x_h^{(i)} - x_h^{(j)}|$ se traducen en fuertes cambios en esa dirección producto de la baja correlación. Asimismo, el exponente p_h está relacionado con la suavidad de la función en la dirección de la coordenada h, donde $p_h = 2$ corresponde a funciones suaves y valores cercanos a 1 corresponden a funciones menos suaves. Nótese

que la correlación del error le permite al modelo de la Ecuación [1] interpolar sobre la data observada y, al mismo tiempo, preservar su continuidad.

Sacks et al. (2) establecen que es posible prescindir del modelo de regresión lineal de la Ecuación 1 y lo reemplazan por una simple constante (μ). Sostienen que esta simplificación no afecta el desempeño de la predicción pues el uso de la correlación de los errores resulta sumamente efectiva, y que su experiencia previa en problemas de ingeniería no sugiere la presencia de fuertes tendencias en la región de interés. Es decir, que suponen que el modelo original es estacionario, lo que conduce al siguiente modelo sustituto conocido como modelo DACE:

$$y\left(\boldsymbol{x}^{(i)}\right) = \mu + \xi\left(\boldsymbol{x}^{(i)}\right) \quad (i = 1,\cdots,n) \qquad [3]$$

El modelo DACE tiene *2k+2* parámetros: $\mu, \sigma^2, \theta_1, \ldots, \theta_k$ y $p_1, \ldots, p_k$, y su estimación consiste en determinar los valores que maximizan la verosimilitud de la muestra. Luego de determinar los parámetros del modelo DACE, la estimación del valor funcional de un nuevo punto se calcula mediante el mejor predictor lineal no sesgado, dado por la siguiente expresión:

$$\hat{y}(x^*) = \hat{\mu} + r' R^{-1}\left(y - \mathbf{1}\hat{\mu}\right) \qquad [4]$$

donde **y** es el vector de dimensión *n* que contiene los valores funcionales de las observaciones, **R** es la matriz de correlación del error, **1** es un vector de dimensión *n* cuyos elementos son unos (1), $\hat{\mu}$ es el valor medio estimado y **r** es un vector de dimensión *n* que contiene las correlaciones entre el error en el punto **x*** donde se está realizando la predicción y los errores en puntos previamente considerados.

Adicionalmente, el modelo DACE provee una estimación del error cuadrático medio del predictor, la cual se calcula como sigue:

$$s^2(\boldsymbol{x}^*) = \sigma^2\left[1 - r' R^{-1} r + \frac{\left(1 - \mathbf{1}' R^{-1} r\right)^2}{\mathbf{1}' R^{-1}\mathbf{1}}\right] \qquad [5]$$

A partir de las Ecuaciones [4 y 5] se puede demostrar que el modelo DACE es interpolante, es decir, una vez que se conoce el valor funcional de una observación, la incertidumbre en ese punto (medida por el error cuadrático medio) es cero y el modelo predice exactamente dicho valor funcional.

3. Algoritmo EGO

EGO (*Efficient Global Optimization*) (1) es un algoritmo de optimización global bayesiana caracterizado por una sólida fundamentación teórica y por ser altamente eficiente en términos de la cantidad requerida de evaluaciones de la función objetivo, por ello es utilizado en la búsqueda de extremos globales de funciones de alto costo computacional. EGO emplea el modelo DACE (2) para el ajuste sucesivo de superficies de aproximación; es decir, una nueva superficie se ajusta cada vez que se incorpora un punto a la muestra. Asimismo, utiliza una Figura de mérito (mejora esperada (7, 1)) que guía el proceso de búsqueda global de forma tal que, además de explorar regiones donde la superficie de aproximación es óptima y aprovechar la eficiencia del predictor DACE, también pone énfasis en el muestreo de regiones inexploradas donde la predicción tiene una incertidumbre considerable.

El algoritmo EGO incluye los siguientes pasos:

1. Se genera una muestra inicial mediante un diseño de experimento de muestreo uniforme con tamaño aproximadamente igual a *10k*, donde *k* es el número de variables. El diseño de experimento empleado en [1] es un hipercubo latino especial que tiene la propiedad de que todas las proyecciones de una y dos dimensiones son cubiertas uniformemente.

2. Luego de evaluar la función objetivo en los puntos iniciales, se asume $p_h=2$ y se estiman los parámetros θ_h del modelo DACE que maximizan la verosimilitud de la muestra. Nótese que μ y σ^2 se calculan en función de θ_h y p_h.
3. Se valida el modelo DACE obtenido en el paso anterior mediante el método de validación cruzada. Si los residuales de la validación cruzada son menores que 3 en magnitud, se dice que el modelo es satisfactorio. En caso contrario, se reajusta el modelo luego de aplicar una transformación logarítmica (*log(y), -log(-y)*) o inversa *(-1/y)* sobre la variable dependiente. Si la aplicación de una transformación resulta en un modelo satisfactorio, se emplea la función transformada en el resto del análisis. En caso de que ninguna de las transformaciones consideradas derive en un modelo válido, no es posible continuar.
4. Se determina el punto donde se maximiza la mejora esperada.
5. Si la mejora esperada es menor que 1% del mejor valor actual de la función (en la escala no transformada) o se ha alcanzado el número máximo de iteraciones, el algoritmo se detiene. En caso contrario, se evalúa la función objetivo en el punto donde es máxima la mejora esperada, se incorpora dicho punto a la muestra, se reestiman los parámetros del modelo DACE y se regresa al paso 4.

4. Algoritmo propuesto

La principal debilidad del algoritmo EGO es suponer que la función objetivo es estacionaria ya que existen muchos problemas de optimización global donde no es posible garantizar tal condición. Ante una función objetivo no estacionaria, EGO puede arrojar resultados erróneos. Otras debilidades importantes de EGO son que su propia naturaleza lo lleva a trabajar con matrices de correlación mal condicionadas y que la reestimación del modelo DACE en cada iteración influye negativamente sobre su tiempo de ejecución. Con el propósito de superar estas debilidades, se ha propuesto un algoritmo, inspirado en EGO, que incluye las mejoras descritas a continuación.

4.1. Modelo aditivo acoplado

EGO (1) utiliza el modelo DACE (2) para el ajuste sucesivo de superficies de aproximación, por lo tanto presupone que la función objetivo es estacionaria, es decir que no experimenta fuertes tendencias en la región de interés. Claramente, no se puede garantizar el cumplimiento de esta suposición ya que existen muchos problemas de optimización global con funciones objetivo no estacionarias. Considerar estacionaria a una función que no lo es puede hacer difícil la identificación de un modelo DACE inicial válido y peor aún, si se identifica uno probablemente no guiará de forma adecuada el proceso de búsqueda global. Es por ello que la utilización de transformaciones para suavizar la respuesta de la función objetivo y lograr un buen ajuste del modelo DACE inicial puede ser visto como un esfuerzo de EGO para satisfacer la condición de modelo estacionario. Adicionalmente, las transformaciones necesariamente fracasan ante funciones que tomen valores positivos y negativos.

Para superar estas dificultades se propone un modelo aditivo de la forma:

$$y(\boldsymbol{x}) = m(\boldsymbol{x}) + err(\boldsymbol{x}) = m(\boldsymbol{x}) + \mu + \xi(\boldsymbol{x}) \qquad [6]$$

donde:

- $m(\boldsymbol{x})$ es un modelo de ajuste suave empleado para capturar la tendencia de la función objetivo. Este modelo debe ser parsimonioso (poca cantidad de parámetros), no interpolante, eficiente y tener un método de identificación de relativo bajo costo computacional (e.g., modelos de regresión lineal y redes neuronales tipo perceptrón multicapa).

– $err(\mathbf{x})$ es el error entre la predicción del modelo $m(\mathbf{x})$ y el verdadero valor de la muestra, el cual es modelado con DACE.

A diferencia de las estrategias tradicionales basadas en modelos aditivos, el modelo de tendencia $m(\mathbf{x})$ se mantiene acoplado al modelo DACE del error durante el progreso del algoritmo. Bajo este escenario, la figura de mérito (mejora esperada) conserva su formulación original aunque su cálculo cambia ligeramente: la incertidumbre del valor de $y(\mathbf{x}^*)$, donde $\mathbf{x}^*$ es una nueva muestra, se modela como una variable aleatoria normal cuyo valor medio es la suma de la predicción del modelo de tendencia y la predicción del error del modelo DACE, y cuya desviación estándar está dada únicamente por el predictor del error cuadrático medio del modelo DACE.

Debido a su alta efectividad y gran popularidad los modelos de tendencia seleccionados son: 1) los modelos de regresión lineal (5), y 2) las redes neuronales de tipo perceptrón multicapa (8). Esto conduce a la definición de las variantes del algoritmo propuesto denominadas: REGO (*Linear Regression - Efficient Global Optimization*) y NEGO (*Neural Network - Efficient Global Optimization*).

REGO considera los siguientes modelos de regresión: lineal simple, cuadrático simple (sin efectos combinados), y cuadrático completo (con efectos combinados). Siguiendo este orden se estiman los parámetros mediante el método de regresión y se calcula el valor de ajuste de R^2 para cada modelo. Si el valor de ajuste de R^2 para un modelo dado es mayor a un valor preestablecido (típicamente 0,7) entonces el modelo es seleccionado y no se consideran los restantes, lo cual favorece a los modelos más simples. Por el contrario, si ninguno de los modelos identificados satisface este criterio se selecciona el que tenga el mayor valor de ajuste de R^2.

Por su parte, NEGO incorpora como modelo de tendencia a una red neuronal tipo perceptrón multicapa con una capa oculta. Con el propósito de obtener una red parsimoniosa, el número de unidades de la capa oculta es determinado según la siguiente regla heurística (9):

$$(e+1) * o + (o+1) * s \leq \psi * k \qquad [7]$$

donde e es el número de unidades de entrada, o es el número de unidades de la capa oculta, s es el número de unidades de salida, k es el tamaño de la muestra y ψ es el factor de parametrización (típicamente $0{,}1 \leq \psi \leq 0{,}5$). Como función de activación se utiliza la tangente hiperbólica y el algoritmo de entrenamiento empleado es el método de retropropagación (10). Puesto que se pretende que la red neuronal sólo capture la tendencia de la función objetivo, no se requiere de un entrenamiento con muchas iteraciones ni con un criterio de parada muy exigente.

NEGO (11) fue propuesto originalmente como una metodología para estimar las distribuciones de permeabilidad y porosidad de yacimientos petroleros a partir de data estática y dinámica disponible. Posteriormente fue empleado en el diseño óptimo de tratamientos de estimulación por fracturamiento hidráulico (12) y en la optimización de procesos SAGD (13).

Finalmente, como medio de diferenciación se denomina XEGO a la variante que, pese a la no adición de un modelo de tendencia ($m(\mathbf{x}) = 0$), incluye el resto de las mejoras del algoritmo EGO propuestas en este trabajo.

4.2. Manejo del mal condicionamiento de la matriz de correlación

El manejo del mal condicionamiento de la matriz de correlación **R** es un aspecto de suma importancia que se deriva de la propia formulación teórica del algoritmo EGO (1). El progreso en la búsqueda de la solución óptima global lleva a colocar puntos cercanos a otros previamente considerados. Cuando dos puntos están muy cercanos sus valores de correlación son casi iguales y las correspondientes columnas de la matriz **R**

son casi idénticas, lo cual origina su mal condicionamiento.

En (1) se calcula la inversa de la matriz de correlación **R** mediante la descomposición en valores singulares. En caso de que dichos valores singulares sean muy pequeños (*e.g.* menores a 10^{-10}) sus recíprocos son igualados a cero. Este método es apropiado para determinar la mejor solución, en términos de mínimos cuadrados, del sistema de ecuaciones lineales $\mathbf{A}.\mathbf{x} = \mathbf{b}$, donde **A** es una matriz cuadrada mal condicionada (14). Sin embargo, se considera inadecuado para el cálculo de la inversa de la matriz de correlación **R**.

Con el propósito de menguar el efecto del mal condicionamiento de la matriz de correlación **R** producto de la colocación de puntos cercanos, se propone la siguiente estrategia:

> Luego de obtener el nuevo punto como resultado de la maximización de la mejora esperada, se determina el Número de Condición (*Nc*) de la matriz de correlación **R**, la cual incluye la columna y fila correspondiente al nuevo punto. Si $Nc>10^8$ entonces la matriz de correlación se considera mal condicionada, en cuyo caso se duplica la distancia entre el nuevo punto y el punto más cercano existente en la muestra. Dicho desplazamiento se realiza sobre la recta definida por ambos puntos. Si la condición de singularidad desaparece el algoritmo prosigue y se evalúa la función objetivo en el punto desplazado. De lo contrario, se repite la operación de desplazamiento. Si se alcanza el número máximo de desplazamientos (típicamente 5) y no ha desaparecido la condición de singularidad, se evalúa el punto original resultante de la maximización de la mejora esperada y se termina el algoritmo.

Esta estrategia provee un resultado analíticamente correcto de la inversa de la matriz de correlación y a la vez, permite el avance del proceso de búsqueda al reducir el grado de incertidumbre de la región correspondiente a la vecindad del punto donde es máximo el valor de la mejora esperada. Esto hace que el algoritmo alcance un balance adecuado entre efectividad y robustez.

Puesto que el algoritmo depende directamente del cálculo de la inversa de la matriz de correlación **R**, si persiste la condición de singularidad de dicha matriz no queda otra alternativa más que detener el proceso de búsqueda. Tal situación puede ocurrir luego de colocar un nuevo punto en una zona suficientemente explorada que aún promete reducir el menor valor funcional identificado hasta el momento. Nótese que esto indica la posible existencia de una mejor solución en este nuevo punto.

4.3. Ajuste ligero del modelo DACE

El algoritmo EGO (1) estima los parámetros del modelo DACE en cada iteración luego de agregar a la muestra un nuevo punto donde se maximiza la mejora esperada. Según nuestra experiencia (11-13) tal reestimación afecta notoriamente el tiempo de ejecución del algoritmo, especialmente en problemas de cinco o más variables. Esto se debe a que en cada iteración se incrementa la dimensión de la matriz de correlación **R**, y por lo tanto, aumenta el costo computacional del cálculo de su inversa empleado para la estimación de los parámetros que maximizan la verosimilitud de la muestra.

Por ello se propone el ajuste ligero del modelo DACE. Dicho ajuste presupone que los parámetros θ_k que maximizan la verosimilitud de la muestra estimados inicialmente, luego de ser ratificados mediante el procedimiento de validación cruzada, capturan efectivamente la naturaleza de la función objetivo. Por lo tanto, en cada iteración, luego de agregar a la muestra el nuevo punto correspondiente al máximo valor de la mejora esperada, únicamente se calcula la inversa de la matriz de correlación **R** y se actualizan los parámetros μ y σ^2.

Indiscutiblemente este esquema de ajuste ligero del modelo DACE ejerce mayor presión en la selección de un diseño de experimento realmente efectivo y la identificación inicial de los parámetros del modelo DACE.

4.4. Descripción general del algoritmo propuesto

El algoritmo propuesto incluye los siguientes pasos:

1. Se genera la muestra inicial mediante un diseño de experimento de muestreo uniforme con tamaño aproximadamente igual a *10k*, donde k es el número de variables.
2. Se identifica el modelo de tendencia según la variante del algoritmo seleccionada (XEGO, REGO o NEGO) de acuerdo a lo expuesto en la Sección 4.1.
3. Se estiman los parámetros del modelo DACE correspondiente al error entre la predicción del modelo de tendencia y el verdadero valor de la función objetivo sobre la muestra inicial (Sección 4.1).
4. Si el modelo DACE del error satisface el criterio de validación cruzada el algoritmo continúa, de lo contrario termina.
5. Se determina el punto donde se maximiza la mejora esperada (Sección 4.1).
6. Se aplica la estrategia para el manejo del mal condicionamiento de la matriz de correlación **R** (Sección 4.2). En caso de no superar dicha condición, se evalúa la función objetivo en el nuevo punto y el algoritmo termina.
7. Se evalúa la función objetivo en el nuevo punto, se incorpora a la muestra y se efectúa el ajuste ligero del modelo DACE del error (Sección 4.3).
8. Si el valor máximo de la mejora esperada relativa es menor a un valor prestablecido (típicamente 1%) o se alcanza el número máximo de iteraciones, el algoritmo se detiene; de lo contrario regresa al paso 5.

5. Estudio de casos y análisis de resultados

Para mostrar la eficacia del algoritmo propuesto, sus variantes (XEGO, REGO y NEGO) fueron codificadas en MATLAB (15) y ejecutadas en una computadora Pentium 4 (1,7 GHz). En particular, XEGO, REGO y NEGO utilizan el algoritmo DIRECT (16, 17) para identificar el modelo DACE inicial (maximizar la verosimilitud de la muestra) y determinar el punto donde se maximiza la mejora esperada. Detalles de su implementación se describen en (18). En todos los casos de estudio las mejoras propuestas produjeron los resultados esperados.

En cuanto a su eficiencia relativa, se compara el desempeño de XEGO, REGO y NEGO frente al programa SPACE (*Stochastic Process Analysis of Computer Experiments*) (3, 4). SPACE es una implementación del algoritmo EGO desarrollada en C++ y se ejecutó en una computadora SUN UltraSPARC III (1,2 GHz). Para realizar la comparación se seleccionó un grupo de funciones objetivo derivables, que han dado resultados satisfactorios en implementaciones de EGO, cuya naturaleza multimodal dificulta en gran medida la identificación de extremos globales (Tabla 1). Estas funciones de prueba son de uso frecuente en la literatura y en paquetes de optimización (1, 16, 17, 19).

En todos los casos la muestra inicial fue generada mediante un hipercubo latino clásico (20). El tamaño de dicha muestra se estableció partiendo de la regla heurística (*10k*) sugerida en (1), aunque hubo casos donde fue necesario aumentar su tamaño para lograr un modelo inicial válido. El algoritmo se detiene cuando la máxima mejora esperada es menor que el 1% del mejor valor funcional obtenido hasta el momento (en la escala no transformada) o cuando alcanza 30 iteraciones. Nótese que SPACE y XEGO requirieron la aplicación de las transforma-

Tabla 1
Funciones de prueba

Caso	Función	Dimensiones	Mínimos Globales
1	Branin	2	3
2	Hock-Shittkowski 5	2	1
3	HGO 468:1	2	1
4	Goldstein-Price	2	1
5	Six-Hump Camel	2	2
6	Hartman 3	3	1
7	Hartman 6	6	1

ciones *ln(y)* y *-ln(-y)* para optimizar las funciones Goldstein-Price (caso 4) y Hartman 6 (caso 7), respectivamente.

Los resultados obtenidos (Tabla 2) revelan que el desempeño de SPACE, XEGO y REGO es similar, excepto en los casos 4 y 7. En el caso 4, SPACE superó a XEGO, REGO y NEGO, aunque éstos arrojaron resultados muy cercanos a la solución óptima global pese a tener un error relativo muy elevado. Esto se atribuye a la naturaleza de la función objetivo Goldstein-Price que presenta dos regiones bien diferenciadas dentro del espacio de búsqueda considerado ($-2 \leq x \leq 2$; $-2 \leq y \leq 2$). La primera con valores funcionales en el orden de los cientos de miles ($\sim 10^5$) que desciende rápidamente hasta un extenso valle (segunda región) con valores en el orden de las decenas, siendo 3 el mínimo global ubicado en (0, -1). Se pudo apreciar que XEGO, REGO y NEGO condujeron de buena manera el proceso de búsqueda pues exploraron más intensamente el valle de la función.

En el caso 7, SPACE y XEGO obtuvieron soluciones similares, aunque XEGO requirió menos evaluaciones de la función objetivo. REGO y NEGO arrojaron resultados bastante alejados del mínimo global pese a haber alcanzado el criterio de parada de la mejora esperada. Esto se atribuye a la escasez de puntos iniciales que imposibilita la identificación de modelos de tendencia y modelo DACE que capturen efectivamente el comportamiento de la función objetivo (Hartman 6). Nótese que sólo se emplearon 72 puntos iniciales, cantidad que supera ligeramente el número de vértices de un hipercubo de 6 dimensiones (64), lo cual es equivalente a identificar un modelo de 2 variables con sólo 3 o 4 muestras.

En general, los resultados obtenidos son similares a los reportados en (1) siendo estos últimos ligeramente mejores. Por otra parte, NEGO mostró cierta tendencia a requerir mayor número de evaluaciones de la función objetivo, en particular en los casos 1 y 5. Aunado a esto, la menor cantidad de parámetros del modelo de regresión lineal y la simplicidad del método empleado para su identificación, convierten a REGO en la primera opción para la solución de problemas de optimización global que requieran el uso de modelos de tendencia.

Con relación a las mejoras propuestas, los resultados obtenidos son los siguientes:

1. El uso de modelos de tendencia resultó ser efectivo en todos salvo uno de los casos, teniendo REGO mejor desempeño que NEGO. Sin embargo, REGO y NEGO mostraron peor desempeño que SPACE y XEGO en los casos que requirieron el uso de transformaciones [4 y 7]. En el caso 4, REGO y NEGO condujeron bien el proceso de búsqueda pero requirieron mayor número de evalua-

Tabla 2
Resultados sobre las funciones objetivo de prueba

	SPACE	**XEGO**	**REGO**	**NEGO**
Función		**Branin-Hoo (Caso 1)**		
NOI	20	20	20	20
EFO	30	30	30	43
ER (%)	0,026	0,85	0,17	0,35
$\lvert x^*\text{-}x\_opt \rvert$	0,0244	0,0519	0,0511	0,0161
Función		**Hock-Shittkowski 5 (Caso 2)**		
NOI	20	20	20	20
EFO	25	24	23	25
ER (%)	0,002	0,07	0,17	0,08
$\lvert x^*\text{-}x\_opt \rvert$	0,0068	0,0315	0,0612	0,0371
Función		**HGO 468:1 (Caso 3)**		
NOI	20	20	20	20
EFO	24	22	22	22
ER (%)	0,00012	0,92	0,18	0,22
$\lvert x^*\text{-}x\_opt \rvert$	2E-06	0,0107	0,0047	0,0023
Función		**Goldstein-Price (Caso 4)**		
NOI	21	25	21	24
EFO	34	39	51	54 (MNI)
ER (%)	0,97	26,56	87,49	21,53
$\lvert x^*\text{-}x\_opt \rvert$	0,013	0,0509	0,1168	0,037
Función		**Six-Hump Camel (Caso 5)**		
NOI	20	25	25	25
EFO	42	44	46	55 (MNI)
ER (%)	0,0002	0,79	1,66	1,69
$\lvert x^*\text{-}x\_opt \rvert$	0,00015	0,0322	0,0512	0,0476
Función		**Hartman 3 (Caso 6)**		
NOI	30	30	30	30
EFO	35	33	33	36
ER (%)	0,094	0,19	0,13	0,11
$\lvert x^*\text{-}x\_opt \rvert$	0,0109	0,0612	0,0702	0,0110
Función		**Hartman 6 (Caso 7)**		
NOI	65	72	72	72
EFO	93	90	90	85
ER (%)	1,56	1,5	10,25	4,08
$\lvert x^*\text{-}x\_opt \rvert$	0,0560	0,0507	1,0007	1,1151

NOI: Número de observaciones iniciales. **EFO**: Evaluaciones de la función objetivo. **ER(%)**: Error relativo porcentual del valor funcional de la solución obtenida con respecto al valor funcional de la solución óptima. $\lvert \mathbf{x^*\text{-}x\_opt} \rvert$: distancia entre la solución obtenida y el mínimo global. **MNI**: Se alcanzó el máximo número de iteraciones (30).

ciones de la función objetivo; mientras que en el caso 7, fueron incapaces de obtener una solución cercana al óptimo global. En consecuencia, pese a ser una alternativa atractiva y prometedora para la optimización de funciones objetivo no estacionarias, aún requiere mayor estudio.

2. XEGO, REGO y NEGO siempre pudieron superar el mal condicionamiento de la matriz de correlación mediante el uso de la estrategia descrita en la Sección 4.2. Con frecuencia, sólo fue necesario realizar un desplazamiento del nuevo punto.

3. Como consecuencia del ajuste ligero del modelo DACE, se observó consistentemente en todos los casos que la duración de la segunda y demás iteraciones fue sólo una fracción del tiempo requerido por la primera (que incluye maximizar la verosimilitud y la mejora esperada, y opcionalmente identificar el modelo de tendencia). Además, a partir de la segunda iteración no fue tan marcado el incremento del tiempo de ejecución de cada iteración.

6. Conclusiones

En este trabajo se ha propuesto una versión mejorada del algoritmo EGO (1) con la finalidad de superar algunas de sus debilidades más importantes. En este sentido, se puede concluir que:

La incorporación de un modelo aditivo acoplado resultó ser efectiva obteniéndose convergencia al óptimo en casi todos los casos de estudio y en consecuencia, promete ser una alternativa eficaz a ser empleada en la solución de problemas de optimización global con funciones objetivo determinísticas, no estacionarias y de alto costo computacional.

La estrategia para el manejo del mal condicionamiento de la matriz de correlación ofrece resultados analíticamente correctos y brinda al algoritmo un balance adecuado entre efectividad y robustez.

A juzgar por los resultados el esquema de ajuste ligero del modelo DACE captura efectivamente la naturaleza de la función objetivo y brinda mayor velocidad de ejecución sin detrimento en la calidad de la respuesta del algoritmo.

Asimismo, se desarrollaron tres variantes que difieren en el modelo de tendencia utilizado: 1) XEGO, no incluye modelo de tendencia, 2) REGO, emplea modelos de regresión lineal, y 3) NEGO, utiliza redes neuronales tipo perceptrón multicapa. Para determinar su eficiencia, se comparó el desempeño de XEGO, REGO y NEGO frente al programa SPACE (3, 4). En la mayoría de los casos SPACE, XEGO y REGO experimentaron un desempeño similar; sin embargo, NEGO mostró cierta tendencia a requerir mayor número de evaluaciones de la función objetivo. Además, los resultados obtenidos son similares a los reportados por Jones et al. (1).

Agradecimiento

Los autores agradecen el apoyo económico suministrado por el Fondo Nacional de Ciencia, Tecnología e Innovación (Venezuela) a través del proyecto G-97003899.

Referencias Bibliográficas

1. JONES D., SCHONLAU M., WELCH W. ***J of Global Optimization*** 13: 455-492, 1998.
2. SACKS J., WELCH W., MITCHELL T., WYNN H. ***Statistical Science*** 4: 409-435, 1989.
3. SCHONLAU M. Computer Experiments and Global Optimization (Ph.D. Thesis). University of Waterloo, Ontario (Canada), pp. 131, 1997.
4. SPACE. Stochastic Process Analysis of Computer Experiments. Schonlau M. URL: http://www.schonlau.net/space.html. 2001.

5. RAO C.R. ***Linear Statistical Inference and Its Application***, 2nd edition, John Wiley & Sons, pp. 294-301, 1973.
6. MATHERON, G. ***Economic Geology*** 58:1246-1266, 1963.
7. MOCKUS J., TIESIS V., ZILINSKAS, A. ***The Application of Bayesian Methods for Seeking Extremun***. ***Toward Global Optimization, Vol. 2***. Dixon, L. y Szego, G. (Editors), North-Holland, Amsterdam, pp. 117-129, 1978.
8. HAYKIN S. ***Neural Networks. A Comprehensive Foundation*** 2nd Edition, Prentice Hall, 1999.
9. DUDA R., HART P., STORK D. ***Pattern Classification*** 2nd Edition, Wiley-Interscience, pp. 306-318, 2000.
10. RUMELHART D., HILTON G., WILLIAMS R. Learning Internal Representations by Error Propagation. ***Parallel Distributed Processing: Explorations in the Microstructures of Cognition*** (Eds. Rumelhart D. E., McClelland), Vol. 1, MIT Press, Cambridge, MA, pp. 318-362, 1986.
11. QUEIPO N.J., PINTOS S., RINCÓN N., CONTRERAS N., COLMENARES J. Surrogate Modeling-based Optimization for the Integration of Static and Dynamic Data into a Reservoir Description (SPE 63065). ***SPE Annual Technical Conference and Exhibition*** Dallas (USA), pp. 1-10, 2000.
12. QUEIPO N., VERDE A., CANELÓN J., PINTOS S. ***J of Petroleum Science and Engineering*** 35(3-4): 151-166, 2002.
13. QUEIPO N., GOICOCHEA, J., PINTOS S. ***J of Petroleum Science and Engineering*** 35(1): 83-93, 2002.
14. PRESS W., FLANNERY B., TEUKOLSKY S., VETTERLING W. ***Numerical Recipes in C. The Art of Scientific Computing*** 2nd Edition, Cambridge University Press, pp. 59-65, 1999.
15. MATLAB. Ver. 5.3. The MathWorks.
16. JONES D., PERTTUNEN C., STUCKMAN B. ***J of Optimization Theory and Applications*** 79: 157-181, 1993.
17. BJÖRKMAN M., HOLMSTRÖM K. Global Optimization using DIRECT Algorithm in Matlab. ***Advanced Modeling and Optimization (Electronic International Journal)*** 2: 17-37, 1999.
18. COLMENARES J. Desarrollo de un Algoritmo Eficiente de Optimización Global Bayesiana y su Integración a un Ambiente de Procesamiento Distribuido (Tesis de Maestría). Universidad del Zulia, Maracaibo (Venezuela), pp. 138, 2001.
19. DIXON L., SZEGO G. ***The Global Optimization Problem: An Introduction. Toward Global Optimization*** (Eds. Dixon, L. y Szego, G.) Vol. 2, North-Holland, Amsterdam, pp. 1-15, 1978.
20. McKAY M., CONOVER W., BECKMAN R. ***Technometrics*** 21: 239-245, 1979.