

Optimización No Lineal Basada en “Enjambre de Partículas”

M. Susana Moreno, Aníbal M. Blanco,

Planta Piloto de Ingeniería Química PLAPIQUI
(Universidad Nacional del Sur - CONICET)
Camino La Carrindanga km. 7 - 8000 Bahía Blanca - Argentina
{smoreno, ablanco}@plapiqui.edu.ar

Resumen. La optimización basada en “enjambre de partículas” (particle swarm optimization) es una estrategia de programación estocástica que ha cobrado gran popularidad en los últimos años debido a sus buenas propiedades de convergencia a óptimos globales y sencillez de implementación. Sin embargo, al igual que otras técnicas de naturaleza evolutiva, su mayor debilidad radica en el tratamiento de las restricciones. En esta contribución se propone una metodología para manipular en forma eficiente las restricciones del problema de optimización que se encuentran habitualmente en problemas típicos de la ingeniería de procesos.

Palabras clave: Optimización No Lineal, Optimización Estocástica, Enjambre de partículas, Manipulación de Restricciones

1 Introducción

La optimización de modelos no lineales es una actividad de gran interés ingenieril y sumamente desafiante desde el punto de vista de la resolución matemática. La no convexidad inherente a muchas de las funciones no lineales presentes típicamente en los modelos, se manifiesta a través de la existencia de múltiples óptimos locales y se potencia con la escala del problema a medida que crece el número de variables y ecuaciones. En términos generales el problema de optimización no lineal puede formularse como:

$$\begin{aligned} & \text{Min}_{\mathbf{x}} f(\mathbf{x}) \\ & \text{s.a} \\ & \mathbf{h}(\mathbf{x})=\mathbf{0} \\ & \mathbf{g}(\mathbf{x})\leq\mathbf{0} \\ & \mathbf{x}^L\leq\mathbf{x}\leq\mathbf{x}^U \end{aligned} \tag{1}$$

En (1), \mathbf{x} representa el conjunto de variables de optimización acotadas entre \mathbf{x}^L y \mathbf{x}^U , $f(\mathbf{x})$ la función objetivo y $\mathbf{h}(\mathbf{x})=\mathbf{0}$ y $\mathbf{g}(\mathbf{x})\leq\mathbf{0}$, las restricciones de igualdad y desigualdad respectivamente.

La optimización de modelos no lineales basados en estrategias determinísticas [1] está muy desarrollada y permite en la actualidad la resolución de modelos de gran escala de manera muy eficiente. Si bien estas técnicas proporcionan un manejo elegante de las restricciones, las soluciones obtenidas dependen fuertemente de los puntos de partida de los algoritmos, incrementado las chances de convergencia a óptimos locales. Por otra parte, si bien la optimización determinística global también posee un grado notable de desarrollo [2], su arquitectura, basada en la exploración de un árbol de decisiones, introduce la complejidad combinatoria lo que a menudo se traduce en altos tiempos de computo aún para problemas de tamaño moderado. Los métodos determinísticos requieren también de la diferenciabilidad de las funciones involucradas lo que los torna inadecuados para modelos con discontinuidades. Finalmente, la matemática numérica asociada a esta rama de la optimización es sumamente sofisticada.

Estas circunstancias han estimulado en los últimos años un importante desarrollo de estrategias de optimización estocásticas para abordar problemas de interés ingenieril [3]. La mayoría de estas estrategias están basadas en la evolución de poblaciones de soluciones de acuerdo a ciertas reglas específicas inspiradas en procesos naturales. Entre las metodologías estocásticas más difundidas se encuentran los algoritmos genéticos, el recocido simulado, las colonias de hormigas y los enjambres de partículas, entre otras.

Estas metodologías han demostrado poseer muy buenas propiedades de convergencia global en problemas multimodales no restringidos y con posibilidad de existencia de discontinuidades. La presencia de restricciones, sin embargo, constituye un aspecto sumamente desafiante para este tipo de estrategias limitando seriamente la eficiencia de la exploración. Si bien se han elaborado un gran número de técnicas para el manejo de restricciones en esquemas estocásticos [4], por ejemplo la penalización de la violación de las restricciones en la función objetivo, su desempeño suele depender altamente del tipo de problema bajo estudio.

En esta contribución se propone una metodología que emplea enjambres de partículas para explorar el espacio no convexo de soluciones, en combinación con rutinas determinísticas eficientes para la resolución de sistemas de ecuaciones algebraicas. De esta forma se persigue conciliar las fortalezas de ambos enfoques: las propiedades exploratorias de espacios no convexos inherentes a las estrategias estocásticas y la eficiencia de las rutinas determinísticas para asegurar la verificación de las restricciones del modelo en una forma eficiente desde el punto de vista computacional.

2 Optimización Basada en Enjambres de Partículas

La metodología de optimización por enjambre de partículas (particle swarm optimization) fue propuesta por Kennedy y Eberhardt en el año 1995 [5]. Dicha metodología está inspirada en el comportamiento social de poblaciones de individuos. Debido a su sencillez de programación y superior capacidad de convergencia a óptimos globales, esta técnica ha ganado una notable popularidad. En los últimos años se han reportado una cantidad abrumadora tanto de aplicaciones en ingeniería y

ciencia, como de sofisticaciones algorítmicas para mejorar su desempeño (ver por ejemplo [6], [7] y [8]).

Básicamente, el algoritmo consiste en actualizar la posición de cada partícula del enjambre ($i=1, \dots, P$) en cada iteración ($k=1, \dots, K$) de acuerdo a la siguiente expresión:

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_k^i \quad (2)$$

donde \mathbf{v}_k representa una tasa de cambio (velocidad) evaluada de acuerdo a la siguiente expresión:

$$\mathbf{v}_{k+1}^i = w \mathbf{v}_k^i + c_1 r_1 (\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2 r_2 (\mathbf{q}_k - \mathbf{x}_k^i) \quad (3)$$

En (3), \mathbf{p}_k^i representa la mejor posición que ha observado cada partícula hasta el momento y \mathbf{q}_k constituye la mejor posición que ha observado la totalidad del enjambre hasta el momento. Las constantes w , c_1 y c_2 constituyen parámetros del algoritmo y r_1 y r_2 son números aleatorios generados entre 0 y 1. Cabe aclarar que el vector que representa las partículas (\mathbf{x}_k^i), así como el vector de velocidades (\mathbf{v}_k^i) y los que almacenan información de la exploración (\mathbf{p}_k^i y \mathbf{q}_k), poseen la dimensión del espacio de búsqueda del problema (1).

Por inspección de las ecuaciones (2) y (3) se observa que, en cada iteración, cada partícula es corregida por un término que involucra una suerte de “memoria individual” y otro que aporta una componente de “memoria global”. Esta realimentación de soluciones “exitosas” hace que la evolución conduzca al enjambre hacia soluciones óptimas, eventualmente al óptimo global.

El ajuste del valor de los parámetros w , c_1 y c_2 permite regular el balance entre la “exploración” del espacio de búsqueda y la “explotación” de las soluciones con mayores probabilidades de resultar óptimas. Otros parámetros importantes son el número de partículas del enjambre (P) y el número máximo de iteraciones (K) del algoritmo, el cual constituye criterio de terminación habitual.

Como se ha mencionado anteriormente, la presencia de restricciones, especialmente las de igualdad ($\mathbf{h}(\mathbf{x})=\mathbf{0}$), afecta significativamente el desempeño de los algoritmos basados en enjambre por lo que se requiere la implementación de estrategias especiales para su tratamiento. En la sección siguiente se describe el mecanismo propuesto en este trabajo para la manipulación de restricciones.

3 Manipulación de Restricciones

El esquema tradicional para incluir las restricciones en estrategias de optimización estocástica implica incluir en la función objetivo un término que penalice fuertemente la violación de las restricciones. Esta metodología es intuitiva y de sencilla implementación pero, en general, resulta sumamente dependiente de los “pesos” asignados a las penalizaciones los cuales suelen ser problema-dependientes.

Por otra parte, la estrategia de descartar directamente las soluciones que no verifican las restricciones no tiene el problema del caso anterior, pero habitualmente

los algoritmos emplean mucho tiempo en este proceso hasta lograr soluciones factibles para dirigir la búsqueda.

En este trabajo proponemos utilizar el modelo de la eq. (1) como una caja negra para el algoritmo de optimización por enjambre de partículas. En otras palabras se trata de optimizar una función objetivo $\Phi(\mathbf{x})$, definida en términos de las restricciones del modelo original:

$$\begin{aligned} & \text{Min}_{\mathbf{x}} \Phi(\mathbf{x}) \\ & \text{s.a} \\ & \Phi(\mathbf{x}) = f\{f(\mathbf{x}), \mathbf{h}(\mathbf{x}), \mathbf{g}(\mathbf{x}), \mathbf{x}^L - \mathbf{x}, \mathbf{x} - \mathbf{x}^U\} \end{aligned} \quad (4)$$

Para el tratamiento subsiguiente asumiremos que \mathbf{x} tiene N elementos, $\mathbf{h}(\cdot)$ tiene H elementos y $\mathbf{g}(\cdot)$ tiene G elementos. También asumimos que las restricciones $\mathbf{h}(\cdot)$ y $\mathbf{g}(\cdot)$ son diferenciables.

La idea básica es dividir el vector de variables de decisión en un vector correspondiente a los grados de libertad, \mathbf{x}_{GL} (de dimensión N-H) y un vector de variables calculadas, \mathbf{x}_{VC} (de dimensión H).

El algoritmo de enjambre de partículas manipula entonces el vector $\mathbf{x} = [\mathbf{x}_{GL}, \mathbf{x}_{VC}^0]$ de acuerdo a las ecuaciones (2) y (3), donde \mathbf{x}_{VC}^0 es el punto de partida para el algoritmo de solución del sistema de ecuaciones $\mathbf{h}(\mathbf{x}_{GL}, \mathbf{x}_{VC}) = \mathbf{0}$. El vector \mathbf{x}_{GL} se mantiene fijo durante la resolución de $\mathbf{h}(\mathbf{x}_{GL}, \mathbf{x}_{VC}) = \mathbf{0}$.

La resolución del sistema $\mathbf{h}(\mathbf{x}_{GL}, \mathbf{x}_{VC}) = \mathbf{0}$ desde el punto \mathbf{x}_{VC}^0 empleando un método determinístico conduce, si existe, a una solución \mathbf{x}_{VC}^* .

Con esta solución se calcula el valor de las restricciones $\mathbf{g}(\mathbf{x}_{GL}, \mathbf{x}_{VC}^*)$, $\mathbf{x}_{VC}^L - \mathbf{x}_{VC}^*$ y $\mathbf{x}_{VC}^* - \mathbf{x}_{VC}^U$. Si ninguna de estas restricciones se viola se procede al cálculo de la función objetivo $f(\mathbf{x}_{GL}, \mathbf{x}_{VC}^*)$ del problema (1).

Cabe aclarar que los valores de los grados de libertad \mathbf{x}_{GL} son parámetros (datos) para todas las funciones ($\mathbf{h}(\cdot)$ y $\mathbf{g}(\cdot)$), al estar proporcionadas por el algoritmo de enjambre de partículas en cada iteración.

Si el sistema de ecuaciones $\mathbf{h}(\mathbf{x}_{GL}, \mathbf{x}_{VC}) = \mathbf{0}$ no posee una solución para la partícula $[\mathbf{x}_{GL}, \mathbf{x}_{VC}^0]$ entonces se asigna un valor infinito para la función objetivo ($\Phi = \infty$), con el objeto de evitar que la partícula visite nuevamente esa posición.

Si el sistema $\mathbf{h}(\mathbf{x}_{GL}, \mathbf{x}_{VC}) = \mathbf{0}$ posee una solución, \mathbf{x}_{VC}^* , para $[\mathbf{x}_{GL}, \mathbf{x}_{VC}^0]$, pero el modelo no es factible con respecto a las restricciones de desigualdad, se devuelve un valor de función objetivo igual a la máxima violación de las restricciones de desigualdad ($\Phi = \max\{\mathbf{g}(\mathbf{x}_{GL}, \mathbf{x}_{VC}^*), \mathbf{x}_{VC}^L - \mathbf{x}_{VC}^*, \mathbf{x}_{VC}^* - \mathbf{x}_{VC}^U\}$).

Finalmente, si la partícula $[\mathbf{x}_{GL}, \mathbf{x}_{VC}^0]$ es factible tanto para las restricciones de igualdad como para las de desigualdad, se devuelve un valor de función objetivo $\Phi = f(\mathbf{x}_{GL}, \mathbf{x}_{VC}^*) - \Phi_{MAX}$, donde Φ_{MAX} es una constante de gran magnitud.

El pseudocódigo de la Fig. 1 resume el procedimiento descrito.

1. *Particionar el vector de variables de optimización en un sub vector de grados de libertad y otro de condiciones iniciales de las variables calculadas: $\mathbf{x}^i = [\mathbf{x}_{GL}^i, \mathbf{x}_{VC}^i]$ ($i=1, \dots, P$)*
2. *Inicializar aleatoriamente el enjambre entre sus cotas*
3. *Para cada partícula calcular la función objetivo Φ de acuerdo al siguiente procedimiento:*
 - a. *Fijar \mathbf{x}_{GL}^i*
 - b. *Resolver el sistema $\mathbf{h}(\mathbf{x}_{GL}^i, \mathbf{x}_{VC}^i) = \mathbf{0}$ desde el punto $\mathbf{x}_{VC}^i = \mathbf{0}$*
 - i. *Si el sistema no posee solución entonces asignar $\Phi = \infty$*
 - ii. *Si el sistema tiene solución, \mathbf{x}_{VC}^i , calcular $\mathbf{g}(\mathbf{x}_{GL}^i, \mathbf{x}_{VC}^i)$, $\mathbf{x}_{VC}^L - \mathbf{x}_{VC}^*$ y $\mathbf{x}_{VC}^* - \mathbf{x}_{VC}^U$*
 1. *Si $\mathbf{g}(\mathbf{x}_{GL}^i, \mathbf{x}_{VC}^*) > \mathbf{0}$ o $\mathbf{x}_{VC}^L - \mathbf{x}_{VC}^* > \mathbf{0}$ o $\mathbf{x}_{VC}^* - \mathbf{x}_{VC}^U > \mathbf{0}$ entonces asignar $\Phi = \max\{\mathbf{g}(\mathbf{x}_{GL}^i, \mathbf{x}_{VC}^*), \mathbf{x}_{VC}^L - \mathbf{x}_{VC}^*, \mathbf{x}_{VC}^* - \mathbf{x}_{VC}^U\}$*
 2. *Si $\mathbf{g}(\mathbf{x}_{GL}^i, \mathbf{x}_{VC}^*) \leq \mathbf{0}$ y $\mathbf{x}_{VC}^L - \mathbf{x}_{VC}^* \leq \mathbf{0}$ y $\mathbf{x}_{VC}^* - \mathbf{x}_{VC}^U \leq \mathbf{0}$ entonces calcular $f(\mathbf{x}_{GL}^i, \mathbf{x}_{VC}^*)$, y asignar $\Phi = f(\mathbf{x}_{GL}^i, \mathbf{x}_{VC}^*) - \Phi_{MAX}$*
4. *Actualizar el enjambre de acuerdo a las ecuaciones (2) y (3)*
5. *Verificar el criterio de terminación $k=K$ (nro. máximo de iteraciones)*
 - a. *Si el criterio de terminación no se verifica ir a 3*
6. *Terminar*

Fig. 1. Pseudocódigo para la evaluación de restricciones

La función objetivo $\Phi(\mathbf{x})$ del modelo (4) constituye una métrica de la calidad del modelo de optimización (1). Si la partícula es infactible respecto de $\mathbf{h}(\cdot)$, la función $\Phi(\mathbf{x})$ asume un valor muy desfavorable. Si la partícula es infactible con respecto de las restricciones de desigualdad ($\mathbf{g}(\cdot)$) y cotas sobre \mathbf{x}_{VC} la función $\Phi(\mathbf{x})$ asume un valor positivo finito. Finalmente, si la partícula es factible, $\Phi(\mathbf{x})$ asume un valor negativo, proporcional al valor de la función objetivo del modelo original $f(\mathbf{x})$. De esta forma se consigue transformar el problema (1), restringido, en una versión no restringida donde una única función objetivo se utiliza para cuantificar tanto la infactibilidad como la optimalidad relativa de las partículas factibles. En la Fig. 2 se esquematiza la definición de la función objetivo del modelo (4).

$\Phi(\mathbf{x}) = \infty$	$\mathbf{h}(\mathbf{x}_{GL}, \mathbf{x}_{VC}) = \mathbf{0}$ no posee solución
↑	
$\Phi(\mathbf{x}) = \max\{\mathbf{g}(\mathbf{x}_{GL}, \mathbf{x}_{VC}^*), \mathbf{x}_{VC}^L - \mathbf{x}_{VC}^*, \mathbf{x}_{VC}^* - \mathbf{x}_{VC}^U\}$	Partículas infactibles
↓	
$\Phi(\mathbf{x}) = 0$	
↓	
$\Phi = f(\mathbf{x}_{GL}, \mathbf{x}_{VC}^*) - \Phi_{MAX}$	Partículas factibles

Fig. 2. Función $\Phi(\mathbf{x})$

La resolución del sistema de ecuaciones $\mathbf{h}(\mathbf{x}_{GL}, \mathbf{x}_{VC}) = \mathbf{0}$ en el paso 3.b se realiza empleando una rutina de resolución de sistemas de ecuaciones algebraicas. En prácticamente todas las plataformas de programación matemática se encuentran

disponibles este tipo de rutinas en versiones que permiten resolver sistemas no lineales de gran escala de manera eficiente y robusta. La mayoría de estas rutinas están basadas en variaciones del método de Newton.

Uno de los requerimientos de la metodología propuesta es la definición de las variables que oficiarán de grados de libertad (\mathbf{x}_{GL}) y de las que serán calculadas a partir del sistema de ecuaciones (\mathbf{x}_{VC}). Dado que los elementos del vector \mathbf{x}_{GL} serán parámetros del sistema $\mathbf{h}(\mathbf{x}_{GL}, \mathbf{x}_{VC})=\mathbf{0}$, es posible seleccionarlos de manera de eliminar muchas de las no linealidades presentes y de esa forma lograr sistemas lineales o moderadamente no lineales.

Finalmente, el parámetro ΦMAX , una constante de gran magnitud restando al valor de $f(\mathbf{x})$, se utiliza para asegurar el sentido de la minimización una vez que se verifica la factibilidad de la solución. Este valor de ΦMAX debe ser mayor que: $\{\text{Max}_x f(\mathbf{x}) \text{ s.a } \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U\}$.

A medida que el enjambre evoluciona, sus individuos tienden a concentrarse en regiones próximas al óptimo, por lo tanto, muchas de las partículas del enjambre final están cerca a la solución deseada. En este trabajo se emplea la partícula \mathbf{p}_K como la representativa de la solución final. Cabe aclarar que para obtener la solución real del modelo (1) es necesario resolver una vez más el sistema de ecuaciones algebraicas para la partícula \mathbf{p}_K dado que el vector correspondiente de variables calculadas es el punto de partida (\mathbf{x}_{VC}^0) de $\mathbf{h}(\mathbf{x})=\mathbf{0}$ y no su solución (\mathbf{x}_{VC}^*).

Una alternativa posible al algoritmo propuesto surge de expresar todas las restricciones de desigualdad $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ como un sistema de restricciones de igualdad mediante la introducción de variables auxiliares ξ , tal que $\mathbf{g}(\mathbf{x}) + \xi = \mathbf{0}$ con $\xi \geq \mathbf{0}$. De esta forma, se consigue eliminar las restricciones $\mathbf{g}(\cdot)$ en el paso 3.b.ii del algoritmo debiéndose verificar únicamente la factibilidad de las variables calculadas (\mathbf{x}_{VC}^* , ξ^*) con respecto de sus cotas. Por supuesto, esto se logra a expensas de incrementar el tamaño del sistema de ecuaciones algebraicas a resolver en el paso 3.b.

En las secciones siguientes se describe la implementación del algoritmo propuesto y se ilustra su desempeño por medio de una serie de problemas típicos de la ingeniería de procesos.

4 Implementación Computacional

El algoritmo descrito en la sección anterior fue implementado en lenguaje Fortran. Se seleccionó esta plataforma esencialmente debido a su capacidad de proporcionar ejecuciones rápidas y por la disponibilidad de una amplia variedad de subrutinas de cálculo, en particular de resolución de sistemas de ecuaciones algebraicas no lineales.

Para todos los ejemplos desarrollados en este trabajo se utilizó una versión estándar del algoritmo de enjambre de partículas con la parametrización detallada en la Tabla 1, la cual se considera apropiada para la mayoría de modelos de optimización no lineal de escala pequeña y moderada.

Para resolver el modelo de ecuaciones algebraicas del paso 3.b se adoptó la subrutina DNEQNF de la librería de rutinas matemáticas IMSL que resuelve un sistema de ecuaciones algebraico no lineal empleando el método de Powell utilizando

una aproximación de diferencias finitas del Jacobiano del sistema.

Table 1. Parametrización del algoritmo de enjambre de partículas.

Parámetro	Descripción	Valor
P	Número de partículas	30
K	Máximo número de iteraciones	250
w	Factor de inercia	1.0
c ₁	Parámetro cognitivo	2.0
c ₂	Parámetro social	2.0

5 Resultados

A continuación se describen los ejemplos utilizados para testear el algoritmo propuesto. Todos ellos han sido extraídos de [9] y constituyen problemas clásicos de ingeniería química, ampliamente utilizados en estudios de optimización no lineal. Si bien se trata de modelos de pequeña escala, constituyen problemas desafiantes por poseer no convexidades y por lo tanto múltiples óptimos locales.

En la Tabla 2 se proporciona para cada ejemplo el porcentaje de soluciones exitosas a partir de 10 experimentos, junto con el tiempo de cómputo promedio en cada caso. Se considera una solución como exitosa, si el valor de la función objetivo logrado se encuentra a un 1% o menos del óptimo global. Las soluciones no exitosas corresponden a otros óptimos (locales) o soluciones factibles sub-óptimas. El tiempo de cómputo se calculó con la subrutina CPU_TIME() de Fortran. Las experiencias se efectuaron en una computadora de escritorio con procesador AMD Athlon(tm) XP 3000+ con 490.864KB RAM.

Ejemplo 1: Optimización del proceso de alquiler

$$\begin{aligned}
 & \text{Min}_x f=5.05x_1+0.035x_2+10x_3+3.36x_5-0.067x_4x_7 \\
 & \text{s.a} \\
 & x_1-1.22x_4+x_5=0 \\
 & x_9+0.222x_{10}-35.82=0 \\
 & 3x_7-x_{10}-133=0 \\
 & 86.35+1.098x_8+0.038x_8^2+0.325(x_6-89)-x_7=0 \\
 & x_6(x_4x_9+1000x_3)-98000x_3=0 \\
 & x_2+x_5-x_1x_8=0 \\
 & -x_1(1.12+0.13167x_8-0.00667x_8^2)+x_4=0 \\
 & x \geq (1.0, 1.0, 0.0, 1.0, 0.0, 85.0, 90.0, 3.0, 1.2, 145.0) \\
 & x \leq (2000.0, 16000.0, 120.0, 5000.0, 2000.0, 93.0, 95.0, 12.0, 4.0, 162.0) \\
 & \text{Solución global} \\
 & x=(1728.3, 16000.0, 98.1, 3056.0, 2000.0, 90.6, 94.2, 10.4, 2.6, 149.6)
 \end{aligned}$$

$$f=-1161.3$$

Este modelo posee tres grados de libertad (10 variables y 7 ecuaciones). Seleccionando las variables x_1 , x_6 y x_9 como elementos del vector de grados de libertad, es posible eliminar varias de las no linealidades presentes en el sistema de ecuaciones algebraicas haciéndolo de sencilla resolución para la rutina calculo de raíces.

Ejemplo 2: Diseño de una red de intercambiadores de calor

$$\begin{aligned} & \text{Min}_x f=x_1+x_2+x_3 \\ & \text{s.a} \\ & 100000(x_4-100)-120x_1(300-x_4)=0 \\ & 100000(x_5-x_4)-80x_2(400-x_5)=0 \\ & 100000(500-x_5)-40x_3(600-500)=0 \\ & x \geq (0.0, 0.0, 0.0, 100.0, 100.0) \\ & x \leq (15843.0, 36250.0, 10000.0, 300.0, 400.0) \end{aligned}$$

$$\begin{aligned} & \text{Solución global} \\ & x=(579.3, 1360.0, 5110.0, 182.0, 295.6) \\ & f=7049.2 \end{aligned}$$

Este modelo posee dos grados de libertad (5 variables y 3 ecuaciones). Seleccionando las variables x_1 y x_2 como elementos del vector de grados de libertad, se consigue eliminar la totalidad de las bilinealidades presentes en el sistema de ecuaciones algebraicas.

Tabla 2. Resultados

Ejemplo	Porcentaje de exito [%]	Tiempo de CPU [s]
1	100	1.18
2	90	0.17
3	70	2.14
4	80	0.19

Ejemplo 3: Síntesis de una red de intercambio de calor

$$\text{Min}_x f = 1200 \left\{ 800 / \left\{ 2.5(2/3[(320-t_2)(300-t_1)]^{0.5} + 1/6[(320-t_2)+(300-t_1)]) \right\} \right\}^{0.6} \\ + 1200 \left\{ 1000 / \left\{ 0.2(2/3[(340-t_4)(300-t_3)]^{0.5} + 1/6[(340-t_4)+(300-t_3)]) \right\} \right\}^{0.6}$$

s.a

$$f_1 + f_2 - 10 = 0$$

$$f_1 + f_6 - f_3 = 0$$

$$f_2 + f_5 - f_4 = 0$$

$$f_3 + f_7 - f_3 = 0$$

$$f_6 + f_8 - f_4 = 0$$

$$100f_1 + t_2f_6 - t_1f_3 = 0$$

$$100f_2 + t_2f_5 - t_3f_4 = 0$$

$$f_3(t_2 - t_1) - 800 = 0$$

$$f_4(t_2 - t_3) - 1000 = 0$$

$$0 \leq f \leq 10$$

$$(100.0, 100.0, 100.0, 100.0) \leq t \leq (290.0, 310.0, 290.0, 330.0)$$

Solución global

$$f = (0.0, 10.0, 10.0, 10.0, 0.0, 10.0, 10.0, 0.0)$$

$$t = (200.0, 280.0, 100.0, 200.0)$$

$$f = 12292.5$$

Este modelo posee tres grados de libertad (12 variables y 9 ecuaciones). Se seleccionaron las variables f_3 , f_4 y t_2 como elementos del vector correspondiente. Este problema posee dos extremos locales además del global, con valores de función objetivo 13380.8 y 15446.9.

Ejemplo 4: Diseño de una red de reactores

$$\text{Min}_x f = -x_4$$

s.a

$$x_1 - 1 + 0.09755988x_1x_5 = 0$$

$$x_2 - x_1 + 0.0966228x_2x_6 = 0$$

$$x_3 + x_1 - 1 + 0.0391908x_3x_5 = 0$$

$$x_4 - x_3 + x_2 - x_1 + 0.0352717x_4x_6 = 0$$

$$x_5^{0.5} + x_6^{0.5} - 4 \leq 0$$

$$x \geq (0.0, 0.0, 0.0, 0.0, 0.0, 0.0)$$

$$x \leq (1.0, 1.0, 1.0, 1.0, 16.0, 16.0)$$

Solución global

$$x = (0.771462, 0.516997, 0.204234, 0.388812, 3.036504, 5.096052)$$

$$f = -0.388812$$

Este modelo posee dos grados de libertad (x_5 y x_6) y dos óptimos locales muy próximos al extremo global con valores de función objetivo -0.3881 y -0.375.

6 Conclusiones y Trabajo Futuro

En este trabajo se propuso un algoritmo de optimización global basado en una estrategia de enjambre de partículas combinada con una rutina determinística para la resolución del sistema de ecuaciones algebraicas. Esta estrategia permite explorar en forma eficiente los sub-espacios de los grados de libertad y de los puntos de partida del sistema de ecuaciones. Seleccionando apropiadamente los grados de libertad, es posible simplificar el sistema de ecuaciones a resolver mediante la eliminación de gran parte de las no linealidades presentes en el modelo original.

Por otra parte, la estrategia adoptada para cuantificar la calidad de las soluciones, elimina la necesidad de implementación de términos de penalización u otras técnicas para evitar la violación de las restricciones.

Cabe aclarar que no se persiguió en este estudio asegurar la convergencia al óptimo global, sino obtener soluciones óptimas o factibles con mínimo esfuerzo de cómputo. Una parametrización “ad hoc” del algoritmo de enjambre o su hibridización con otras técnicas, posibilitaría una mejora desde el punto de vista de la optimización global.

La eficiencia de la implementación en lo que respecta a velocidad de cómputo sugiere la posibilidad de aplicar la metodología desarrollada a sistemas de mayores dimensiones que los habitualmente empleados en estudios teóricos de optimización no lineal, permitiendo abordar problemas de escala industrial [10], [11], [12].

También se pretende extender la metodología para incorporar el tratamiento de variables binarias y posibilitar la resolución de modelos de optimización mixto-entero no lineales [6].

Referencias

1. Biegler, L.T.: Nonlinear Programming-Concepts, Algorithms and Applications to Chemical Engineering. SIAM, Philadelphia (2010).
2. Tawarmalani, M., Sahinidis, N. V.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, Kluwer Academic Publishers, Dordrecht, Vol. 65 in “Nonconvex Optimization and Its Applications” series.3 (2002)
3. Rangaiah, G. P. (editor): Stochastic Global Optimization: Techniques and Applications in Chemical Engineering, Word Scientific Publishing (2010)
4. Coello Coello, C. A.: Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art, Comput. Methods Appl. Mech. Engrg., 19, 1245-1287 (2002)
5. Kennedy, J., Eberhardt. R. C.: Particle Swarm Optimization. In proceedings of IEEE International Conference in Neural Networks, 1942-1948 (1995)
6. Yiqing, L., Xigang Y., Yongjian L.: An Improved PSO Algorithm for Solving NLP/MINLP Problems with Equality Constraints, Comp. Chem. Eng., 31, 153-162 (2007)

7. Chen, T., Chi T.: On the Improvements of the Particle Swarm Optimization Algorithm, *Advances in Engineering Software*, 41, 229-239 (2010)
8. Thangaraj. R. Pant, M., Abraham A., Bouvry P.: Particle Swarm Optimization: Hybridization Perspectives and Experimental Illustrations, *Applied Mathematics and Computation*, 217, 5208-5226 (2011)
9. Ryoo, H.S., Sahinidis, N. V.: Global Optimization of Nonconvex NLPs and MINLPs, *Comp. Chem. Eng.*, 19 (5) 551-566 (1995)
10. Adams, T. A., Seider, W. D.: Practical Optimization of Complex Chemical Processes with Tight Constraints, *Comp. Chem. Eng.*, 32, 2099-2112 (2008)
11. Faber R., Jockenhovel T., Tsatsaronis G.: Dynamic Optimization with Simulated Annealing, *Comp. Chem. Eng.* 29, 273-290 (2005)
12. Egea. J. A., Vries, D., Alonso A. A., Banga J. R.: Global Optimization for Integrated Design and Control of Computationally Expensive Process Models, *Ind. Eng. Chem. Res.* 46 (26) 9148-9157 (2007)