

DESEMPEÑO DE UN ALGORITMO ESPECTRAL PARA OPTIMIZACIÓN RIGUROSA DE PLANTAS DE PROCESOS INDUSTRIALES

**Alejandro O. Domancich^{*}, Juan I. Ardenghi^{*},
Gustavo E. Vazquez^{**†} y Nélica B. Brignole^{**†}**

^{*} Grupo de Investigación y Desarrollo en Computación Científica (GIDeCC)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur - Av. Alem 1253 – 8000 - Bahía Blanca – ARGENTINA
e-mail: alejandrodomancich@hotmail.com;
ardenghi@criba.edu.ar, gvazquez@criba.edu.ar; dybrigno@criba.edu.ar

[†] Planta Piloto de Ingeniería Química (UNS-CONICET)
Complejo CRIBABB Cno. La Carrindanga km 7 - 8000 - Bahía Blanca – ARGENTINA
Tel.: 486 1700 - Fax: 486 1600

Palabras clave: Optimización No Lineal Restringida, Gradiente Espectral, Convergencia Global, Ingeniería de Procesos

Resumen. *Se analizó el desempeño de un nuevo algoritmo basado en la técnica del gradiente espectral proyectado para la optimización del diseño y las condiciones operativas de plantas de procesos industriales. En el presente trabajo se presentan los resultados principales obtenidos al emplear el método espectral en la optimización de problemas académicos y reales, y se compara el desempeño de esta técnica con el de otros algoritmos de optimización comúnmente aplicados en ingeniería de procesos. Los casos industriales analizados incluyen la optimización económica de una planta de producción de benceno, el diseño óptimo de las características geométricas de intercambiadores de calor con el objeto de maximizar el intercambio de energía y la maximización de las ganancias para una planta de producción de cloruro de etilo.*

1 INTRODUCCIÓN

Una característica distintiva de los modelos matemáticos que surgen en problemas de optimización en ingeniería de procesos es su fuerte no linealidad y la presencia de restricciones. En general, los sistemas de ecuaciones resultantes tienden a aumentar en complejidad y tamaño a medida que las formulaciones se vuelven más realistas. El estado del arte en optimización numérica muestra un gran avance en las técnicas empleadas para resolver este tipo de problemas. Existen algoritmos con un soporte teórico muy grande, que los hacen fuertes y confiables a la hora de su implementación. Sin embargo, los problemas que el ingeniero pretende resolver también se han vuelto cada vez más grandes y complejos, pues actualmente es normal que se pretenda modelar en forma rigurosa plantas completas con miles de variables de estado. Por ende, tanto el refinamiento de los procedimientos de software existentes como el diseño de nuevos algoritmos que soporten mayores niveles de complejidad con costos computacionales mínimos constituyen temas vigentes que representan un interesante desafío.

En este trabajo se aplicó el método del Gradiente Espectral Proyectado (SPG)¹ para la optimización de varios casos académicos e industriales, con el objeto de verificar su desempeño en problemas cuyos modelos de simulación exhibieran distintos niveles de complejidad. El método de optimización analizado es una estrategia no monótona de implementación eficiente en cuanto a tiempos de ejecución. En cada iteración, el SPG se combina con una condición para producir una búsqueda lineal no monótona. En el caso de funciones no cuadráticas, el método es asociado a una técnica de globalización que permite garantizar la convergencia a puntos óptimos desde cualquier valor inicial. El método globalizado es muy superior a recientes implementaciones del clásico método de gradientes conjugados, tanto en número de evaluaciones del gradiente como en términos del tiempo de CPU^{1, 2, 3}.

2 EL ALGORITMO DEL GRADIENTE ESPECTRAL

El algoritmo del Gradiente Espectral Proyectado (SPG) está diseñado para resolver problemas de la forma:

$$\begin{cases} \min & f(x) \\ \text{s.a.} & x \in S \end{cases} \quad (1)$$

El SPG fue introducido por Raydan¹ y utiliza direcciones de descenso. En contraste con el clásico método de Cauchy⁴, la elección del paso se realiza de una manera diferente. La forma de escoger la longitud del paso, originalmente presentada por Barzilai y Borwein y descrita en Raydan⁵, requiere menos trabajo computacional y acelera notablemente la velocidad de convergencia. Esta variante combinada con el esquema de búsqueda lineal no monótona desarrollada por Grippo *et al.*⁶ aseguran la convergencia global del SPG.

El método globalizado esta definido por:

$$x_{k+1} = x_k - \frac{1}{\alpha_k} g_k \quad (2)$$

donde g_k es el gradiente de f en x_k , y el escalar α_k viene dado por:

$$\alpha_k = \frac{s_{k-1}^t y_{k-1}}{s_{k-1}^t s_{k-1}} \quad (3)$$

Esta variante involucra sólo operaciones de producto matriz vector mas la evaluación del gradiente, lo que hace que su costo computacional sea considerablemente menor. Detalles teóricos sobre este algoritmo y resultados de pruebas preliminares sobre conjuntos de problemas estándar genéricos que han evidenciado sus bondades pueden consultarse en Raydan¹ y Birgin *et al.*^{2,7}

A continuación se describe el algoritmo completo del método SPG. Dado $z \in R^n$, se define $P(z)$ como la proyección ortogonal de z en S . Si notamos $g(x) = \nabla f(x)$, dados: $\lambda_{\min} > 0, \lambda_{\max} > \lambda_{\min}, \gamma \in (0,1), 0 < \sigma_1 < \sigma_2 < 1$ y un número natural M , a partir de $\lambda_0 \in [\lambda_{\min}, \lambda_{\max}]$ se obtiene $\|P(x_k - g(x_k) - x_k)\|_2 = 0$ mediante el siguiente algoritmo:

Algoritmo SPG

k=0

MIENTRAS $\|P(x_k - g(x_k) - x_k)\|_2 \neq 0$

$$x_+ = P(x_k - \lambda_k g(x_k))$$

$$\alpha = \lambda_k$$

MIENTRAS $\left(f(x_+) > \max_{0 \leq j \leq \min\{k, M-1\}} \{f(x_{k-j})\} + \gamma \langle x_+ - x_k, g(x_k) \rangle \right)$

definir $\alpha \in [\sigma_1 \alpha, \sigma_2 \alpha]$

$$x_+ = P(x_k - \alpha g(x_k))$$

Fin

$$x_{k+1} = x_+$$

$$s_k = x_{k+1} - x_k, \quad y_k = g(x_{k+1}) - g(x_k), \quad b_k = \langle s_k, y_k \rangle$$

$$\text{Si } b_k \leq 0 \quad \lambda_{k+1} = \lambda_{\max}$$

$$\text{Si no } \lambda_{k+1} = \min \left\{ \lambda_{\max}, \max \left\{ \lambda_{\min}, \frac{\langle s_k, b_k \rangle}{b_k} \right\} \right\}$$

$$k = k + 1$$

FIN

3 CASOS DE ESTUDIO

Para analizar el desempeño del SPG se seleccionó el conjunto de problemas de prueba que se describe a continuación, el cual incluye ejemplos académicos e industriales.

3.1 Problemas Académicos

Se analizaron 15 casos de estudio extraídos de la batería de problemas diseñada por Hock y Schittkowitz⁸. Todos los problemas elegidos poseen restricciones de igualdad y sus características generales se resumen en la Tabla 1.

Problema	Dimensión	Nº de Restricciones	Función Objetivo	Restricciones	Solución	Cotas
26	3	1	Polinomial	Polinomial	Teórica	0
27	3	1	Polinomial	Cuadráticas	Teórica	0
28	3	1	Cuadrática	Lineales	Teórica	0
46	5	2	Polinomial	General	Teórica	0
47	5	3	Polinomial	Polinomial	Teórica	0
48	5	2	General	Lineales	Teórica	0
49	5	2	Polinomial	Lineales	Teórica	0
50	5	3	Polinomial	Lineales	Teórica	0
51	5	3	Cuadrática	Lineales	Teórica	0
52	5	3	Cuadrática	Lineales	Teórica	0
53	5	3	Cuadrática	Lineales	Teórica	10
55	6	6	Cuadrática	Lineales	Teórica	8
60	3	1	Polinomial	Polinomial	Práctica	6
77	5	2	Polinomial	General	Práctica	0
79	5	3	Polinomial	Polinomial	Práctica	0

Tabla 1: Lista condensada de los ejemplos académicos considerados en este estudio

La primera columna de esta tabla corresponde al código de problema asignado en Hock y Schittkowitz⁸; la segunda contiene la cantidad de variables de optimización del problema; la tercera es su número de restricciones y las dos siguientes contienen, respectivamente, las características de la función objetivo y de las restricciones. La sexta columna indica si la solución se conoce en forma exacta (problema teórico) o si, por el contrario, sólo se cuenta con soluciones numéricas (problema práctico). Finalmente, la última columna muestra la cantidad de cotas que se imponen sobre las variables.

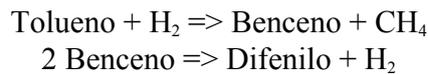
3.2 Aplicaciones Industriales

En esta sección se presentan los tres problemas industriales que se consideraron en este trabajo. Para cada uno de los casos se plantearon modelos matemáticos y se simuló el comportamiento del proceso en estado estacionario. A continuación, cada problema fue optimizado con herramientas computacionales comúnmente utilizadas en ingeniería de

procesos (HYSYS⁹ y GAMS¹⁰), para poder comparar los resultados y los tiempos de cómputo con respecto a los logrados mediante el empleo del método SPG.

3.2.1 Producción de benceno por hidrodealquilación de tolueno (Problema HDA)

En este ejemplo se consideró un modelo de un proceso de hidroalquilación de tolueno para producir benceno. Las reacciones de interés son:



Ambas reacciones se llevan a cabo a una presión de 500 psia, en un rango de temperaturas que va desde 1150°F hasta 1300°F. Estas cotas se imponen por cuestiones operativas ya que por debajo de 1150°F la velocidad de reacción es muy lenta, mientras que por encima de 1300°F se produce una significativa cantidad de hidrocracking. Asimismo se requiere un exceso de hidrógeno (en relación 5 a 1) para prevenir la formación de coque, y los gases efluentes del reactor deben ser rápidamente llevados a 1150°F para que no ocurra el mismo efecto en el intercambiador de calor aguas abajo del reactor. En la Figura 1 se muestra un esquema simplificado del proceso. Las dos corrientes de alimentación, que contienen principalmente tolueno e hidrógeno se calientan y mezclan con dos corrientes de reciclo, antes de ser alimentadas al reactor. La corriente de producto que abandona el reactor contiene hidrógeno (H₂), metano (CH₄), benceno, tolueno y difenilo. Luego, mediante un sistema de separación se logra obtener el benceno (producto principal) y el difenilo (producto secundario). Es importante resaltar que se debe realizar una purga de gases antes de que estos ingresen nuevamente al reactor como reciclo, ya que la acumulación de inertes (CH₄) complicaría el correcto funcionamiento del reactor.

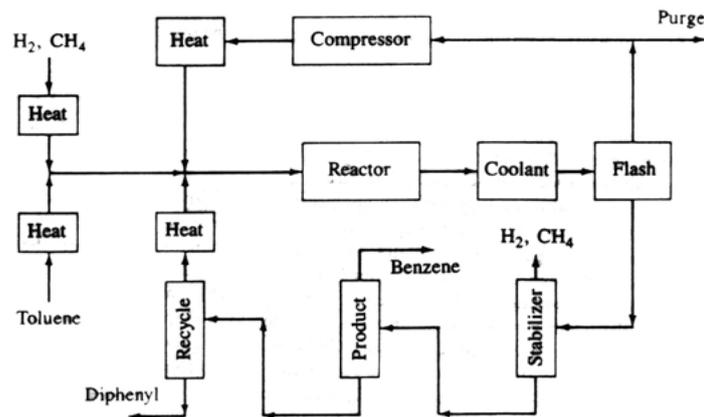


Figura 1: Esquema simplificado de una planta de producción de benceno

Para este problema de optimización se fijó como función objetivo el potencial económico obtenido de la venta del benceno, el difenilo y los gases purgados (combustible). Las

variables de optimización son la conversión que se obtiene en el reactor y la fracción de hidrógeno en la purga. El valor más conveniente para cada una de estas variables es calculado por el algoritmo de optimización. El sistema resultante es de 17 ecuaciones, las cuales provienen de realizar los balances de masa globales y por componentes en los mezcladores, el reactor y el separador (suponiendo separación perfecta).

3.2.2 Red de intercambiadores de calor (Problema RED)

Este caso representa una red de intercambiadores de calor, en los que una corriente principal compuesta por metano, etano, propano y butano, es enfriada de 20°F a -24°F. Para lograr esto se utiliza una red de intercambiadores de calor con servicios externos, dividida en dos sectores en paralelo: uno que contiene un sólo intercambiador y otro que contiene dos. La alimentación principal se divide entonces en dos, para juntarse luego a la temperatura deseada. El diagrama del proceso se muestra en la Figura 2.

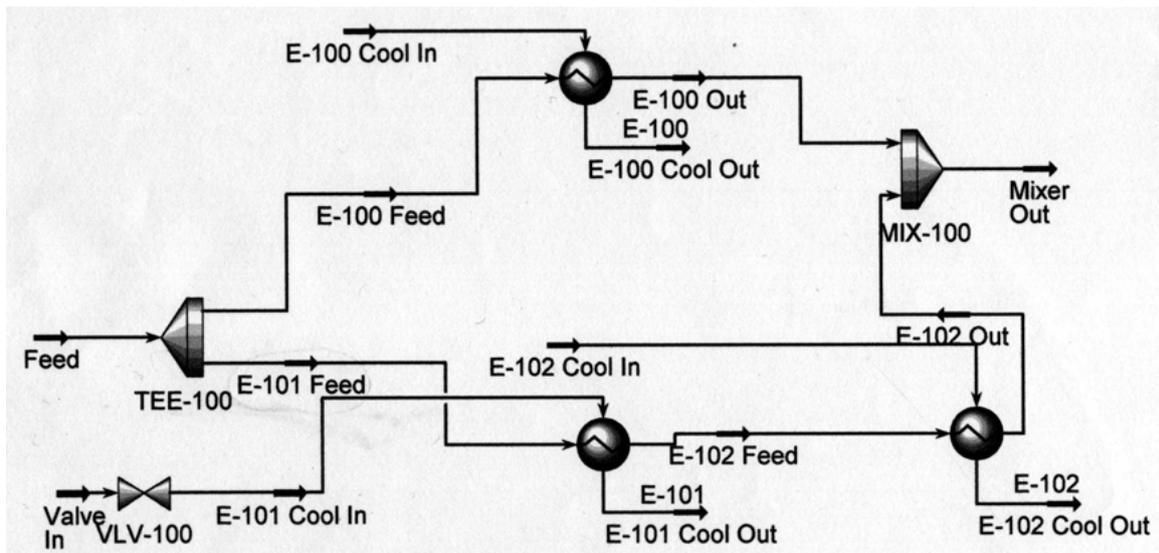
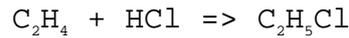


Figura 2: Diagrama de una red de intercambiadores de calor

Este problema es relativamente más complejo que el anterior, ya que posee un número mayor de ecuaciones, las cuales resultan de realizar los balances de masa en los mezcladores y divisores, y los balances de masa y energía en cada uno de los tres intercambiadores de calor. El número total de ecuaciones del sistema es de 61. La función objetivo es la sumatoria de las áreas efectivas de transferencia de los equipos de intercambio de calor (UA), la cual debe ser minimizada para reducir al máximo los costos de instalación de los intercambiadores. La variable de optimización es el caudal denominado E-101 Feed en la Figura 2 que circula por la serie de dos intercambiadores, cuyo valor más conveniente se calcula mediante la optimización del proceso.

3.2.3 Producción de cloruro de etilo (Problema CLE)

Uno de los caminos para producir cloruro de etilo (C_2H_5Cl) es mediante la siguiente reacción de cloruro de hidrógeno (HCl) en fase gas con etileno (C_2H_4), sobre un catalizador de cobre en sílica:



En el proceso mostrado en la Figura 3, la corriente de alimentación está compuesta por 50% molar de HCl, 48% molar de C_2H_4 y 2% molar de N_2 circulando a razón de 100 kmol/hr. y 1 atm. de presión. Como la reacción alcanza solo un 90% de conversión, el cloruro de etilo producido es separado de los reactivos, los cuales son luego reciclados. La separación se logra mediante una columna de destilación, asumiendo que se logra alcanzar una separación perfecta. Para prevenir la acumulación de inertes en el sistema se realiza una purga de los gases que luego se reciclan al reactor. El efecto que produce la magnitud de la purga en el reciclo y en la composición en la alimentación al reactor es de especial interés.

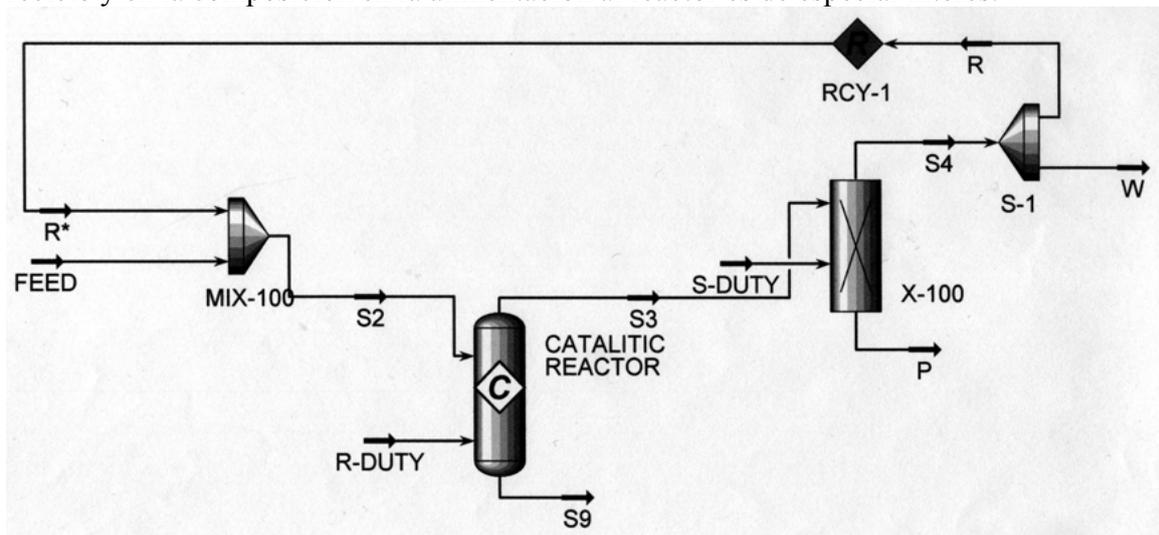


Figura 3: Esquema simplificado de una planta de cloruro de etilo

El sistema resultante de plantear los balances de masa global y por componentes para cada uno de los equipos en estado estacionario es de 32 ecuaciones algebraicas. A pesar de poseer menos ecuaciones que el modelo del problema RED, este sistema es particularmente complicado de simular con HYSYS por el hecho de que contiene un reciclo, lo que obliga al ingeniero a proveer al algoritmo de resolución algunos valores de puntos iniciales razonables, de manera tal de que este pueda llegar al resultado óptimo deseado. La función objetivo a optimizar es el potencial económico resultante de la venta del producto principal, teniendo en cuenta los costos de materia prima y de instalación del reactor (el cual varía de acuerdo al tamaño del equipo). La variable de optimización del proceso es el caudal de purga (W).

4 ANÁLISIS DE DESEMPEÑO

Todos los problemas de optimización descritos en la sección 3 fueron resueltos tanto con el nuevo algoritmo SPG como con dos métodos alternativos CONOPT y MINOS5 (ambos provistos por la plataforma de modelado GAMS¹⁰). CONOPT utiliza un método del tipo gradiente reducido (GRG) luego de efectuar un preprocesamiento del problema, mientras que MINOS5 linealiza las restricciones para luego optimizar mediante un Lagrangiano aumentado. Las aplicaciones industriales también fueron optimizadas con HYSYS⁹, lográndose, en general, arribar a las mismas soluciones que con SPG. A diferencia de la plataforma GAMS, donde el usuario ingresa todas las ecuaciones de modelado, el simulador HYSYS genera sus propios modelos de los equipos de procesos como “caja negra”, y no permite visualizar las ecuaciones que ha utilizado para sus cálculos. Por lo tanto, con el objeto de efectuar comparaciones justas de corridas que correspondan exactamente a los mismos modelos matemáticos de partida, las tablas que se muestran a continuación se elaboraron contrastando resultados de SPG frente a GAMS. Todos los problemas fueron resueltos en una PC Pentium MMX Intel de 233Mhz y 64 Mb, utilizando una tolerancia de 10^{-9} como criterio de cero para la norma de la función objetivo.

La Tabla 2 muestra el desempeño de los algoritmos SPG, CONOPT y MINOS5 en términos de la cantidad de iteraciones y el tiempo de CPU requeridos por cada uno de los métodos para alcanzar convergencia.

Prob. N°	SPG		CONOPT		MINOS5	
	N° Iteraciones	CpuTime (s)	N° Iteraciones	CpuTime (s)	N° Iteraciones	CpuTime (s)
26	83	0.27	17	0.22	79	0.263
27	41	0.53	25	0.83	75	0.293
28	9	0.82	7	0.11	3	0.28
46	475	1.12	42	0.38	95	0.229
47	150	0.28	28	0.285	39	0.28
48	13	0.32	8	0.28	7	0.295
49	708	2.23	22	0.288	21	0.28
50	28	0.71	7	0.11	5	0.269
51	10	0.15	7	0.06	5	0.27
52	9	0.16	7	0.11	5	0.28
53	13	0.16	9	0.17	5	0.276
55	350	1.31	7	0.298	2	0.272
60	20	0.17	---	(*)	---	(*)
77	190	0.26	20	0.306	---	(*)
79	148	0.2	8	0.28	---	(*)
HDA	23	0.88	59	0.28	280	0.36
RED	2	0.39	6	0.16	22	0.31
CLE	159	0.56	19	0.3	14	0.29

Tabla 2: Desempeño de SPG frente a los algoritmos clásicos CONOPT y MINOS5.

(*) El método no alcanza una solución factible.

Se observa que el algoritmo SPG es robusto, pues alcanzó soluciones satisfactorias en todos los casos de estudio, incluso para los problemas N° 60, 77 y 79, que no habían podido ser resueltos con GAMS. Sin embargo, SPG utiliza un número mucho mayor de iteraciones que CONOPT y MINOS5. Esto puede deberse al hecho de que, a diferencia de SPG, estos dos últimos métodos preprocesan el problema antes de efectuar cada iteración, lo cual tiende a reducir el número total de iteraciones requeridas para alcanzar convergencia, a expensas de un aumento en el costo por iteración. Esto es muy evidente, por ejemplo, en los problemas N° 47, 77 y 79, donde SPG tardó menos tiempo que CONOPT en encontrar la solución a pesar de haber requerido muchas más iteraciones. De todos modos, como la reducción en cantidad de iteraciones es en general muy significativa, CONOPT o MINOS5 muchas veces logran mejores tiempos que SPG, como se observa en los problemas N° 46 y 49.

Por otro lado, debe tenerse en cuenta que el número de condición de la matriz Hessiana en el punto óptimo tiene influencia directa en la cantidad de iteraciones y, por ende, en la velocidad de convergencia de SPG. Los casos testigos son los problemas N° 46 y 49, cuyos números de condición son del orden de 10^8 y 10^{11} , respectivamente, y el problema N° 55, cuyo Hessiano es singular. En los tres casos el número de iteraciones requeridas por SPG superó las 300.

La Tabla 3 muestra la razón entre el número de iteraciones y el tiempo de CPU requeridos para lograr convergencia. Este valor es generalmente mucho mayor para SPG en comparación con CONOPT, lo cual implica que el costo por iteración de SPG tiende a ser más bajo que con GAMS.

Probi N°	SPG	CONOPT	MINOS5
26	307.41	77.273	300.3802
27	77.358	30.12	255.9727
28	10.976	63.636	10.71429
46	424.11	110.53	414.8472
47	535.71	98.246	139.2857
48	40.625	28.571	23.72881
49	317.49	76.389	75
50	39.437	63.636	18.58736
51	66.667	116.67	18.51852
52	56.25	63.636	17.85714
53	81.25	52.941	18.11594
55	267.18	23.49	7.352941
60	117.65	---	---
77	730.77	65.359	---
79	740	3.5714	---
HAD	26.136	210.71	777.7778
RED	7.6923	37.5	70.96774
CLE	283.93	63.333	48.27586

Tabla 3. Razón número de iteraciones/tiempo

Una de las características más atractivas de SPG es su rapidez para realizar cada una de sus iteraciones, lo cual representaría una gran ventaja del algoritmo para resolver problemas de gran escala. Sin embargo, debería subsanarse el problema de que, para alcanzar el óptimo, SPG comparativamente requiere un gran número de iteraciones. Dado que la cantidad de evaluaciones de función en este algoritmo es proporcional al número de iteraciones realizadas, el método podría volverse ineficiente en problemas donde la evaluación de la función objetivo o las restricciones fuera muy costosa, características usuales en numerosos problemas de optimización del área de ingeniería de procesos.

La alternativa más natural para superar esta falencia sería la aplicación de un preconditionador adecuado que apunte a reducir la cantidad de iteraciones requeridas por el SPG para lograr convergencia. La disminución en el número de iteraciones siempre conlleva una reducción en el número de evaluaciones de función, lo cual tendría un impacto considerable en los tiempos de ejecución. El procesamiento de los problemas antes o durante su resolución han mostrado ser una herramienta poderosa para la aceleración de ciertos algoritmos, por ejemplo en el caso de CONOPT, donde la efectividad de preprocesamiento que efectúa antes de comenzar a iterar queda clara en la Tabla 2.

5 CONCLUSIONES

En este trabajo quedaron evidenciadas algunas de las bondades y falencias del algoritmo SPG aplicado a problemas académicos e industriales. El método es sencillo de implementar, utiliza pocas operaciones aritméticas y converge globalmente. Fue concebido para resolver problemas de gran porte, lo cual lo ubica como una herramienta potencialmente muy importante para resolver problemas en ingeniería de procesos.

Los resultados numéricos muestran que se alcanza el óptimo con un bajo costo por iteración, pero utilizando mucha cantidad de iteraciones. Esto permite vislumbrar que el siguiente paso en el refinamiento del algoritmo debe orientarse hacia la incorporación de preconditionadores que mejoren el condicionamiento numérico de la matriz Hessiana y aceleren la convergencia del SPG. Existen escasos antecedentes del uso de SPG con preconditionadores^{11,12}. Debería profundizarse el estudio de estas variantes con el objeto de definir una estrategia de preprocesamiento adecuada a las características de los problemas de ingeniería de procesos, lo cual constituye una interesante línea de trabajo futuro.

Los problemas de optimización considerados en este trabajo incluyeron casos de estudio con función objetivo y restricciones no lineales, eligiéndose problemas con restricciones de igualdad. En tal sentido, se proyecta ampliar el espectro de problemas a analizar abordando ejemplos con restricciones de desigualdad, dado que esta característica aparece con frecuencia en la formulación de numerosos problemas del campo de la ingeniería de procesos.

6 REFERENCIAS

- [1] M. Raydan, “The Barzilai and Borwein gradient method for the large-scale unconstrained minimization problem”. *SIAM Journal on Optimization.*, **7**, 1, 26-33, (1997).
- [2] E. Birgin, M. Martinez and M. Raydan, “Non-monotone spectral projected gradient method on convex sets”, *SIAM Journal on Optimization*, **10**, 1196-1212, (2000).
- [3] J. Ardenghi, M. Gibelli and C. Maciel, “The spectral gradient method and unconstrained optimal-control problems” *SIAM Journal on Control and Optimization*, en revisión, (2003).
- [4] A. Cauchy, “Méthodes générales pour la résolution des systèmes d’équations simultanées”, *C.R. Acad. Sci. Par.*, **25**, 536-538, (1947).
- [5] M. Raydan, “On the Barzilai and Borwein choice of step-length for the gradient method”. *IMA Journal on Numerical Analysis*, **13**, 321-326,(1993).
- [6] L. Grippo, F. Lampariello, and S Lucidi, “ A non-monotone line-search technique for Newton’s method”, *SIAM Journal on Numerical Analysis*, **23**, 707-716, (1986).
- [7] E. Birgin, M. Martinez and M. Raydan, “Algorithm 813: SPG-Software for convex constrained optimization”, *ACM Transactions on Mathematical Software*, **27**, 340-349, (2001).
- [8] W. Hock and K. Schittkowski, “Test examples for nonlinear programming codes”, *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin, (1981).
- [9] HYSYS Homepage <http://www.hyprotech.com>
- [10] GAMS Homepage: <http://www.gams.com>
- [11] B.Molina and M.Raydan “ Preconditioned Barzilai–Borwein method for the numerical solution of partial differential equations”, *Numerical Algorithms*, **13**, 45-60, (1996).
- [12] J.L. Lamotte, B. Molina and M.Raydan “Smooth and adaptive gradient method with retards”, *Mathematical and Computer Modeling*, **36**, 1161-1168, (2002).