

Algoritmo evolutivo con un operador genético sencillo aplicado al cómputo de equilibrio de fases en sistemas termodinámicos

Nilda M. Pérez Otero¹, José Luis Zacur¹

¹Facultad de Ingeniería, Universidad Nacional de Jujuy,
S. S. de Jujuy, Argentina
{nilperez, jlzacur}@fi.unju.edu.ar

Resumen. En este trabajo se presenta un algoritmo evolutivo con un operador genético sencillo para la optimización de funciones continuas. Luego de validado, este algoritmo se aplicó en la minimización de la función de Gibbs, función termodinámica relacionada al equilibrio de fases. Finalmente se comparó el desempeño del algoritmo con un algoritmo de búsqueda tabú reportado en la literatura, empleado para el mismo fin. Tanto la validación con las funciones propuestas como *benchmarks* como la contrastación con los resultados del algoritmo tabú se consideran satisfactorias.

Palabras clave: algoritmos evolutivos, optimización, equilibrio de fases, función de Gibbs.

1 Introducción

El cálculo de equilibrio de fase juega un rol crucial en la simulación, diseño y optimización de procesos industriales y, en consecuencia, es esencial en la ingeniería de procesos. Este cálculo puede revelar las composiciones en equilibrio entre fases, llevando a determinar la factibilidad de obtención de productos de interés en procesos de separación. En el sistema físico es importante considerar la distribución de los componentes individuales entre las fases que componen este sistema y la variación de estas cantidades con la presión, la temperatura y la composición. La principal dificultad de estos cálculos radica en el tratamiento de funciones termodinámicas altamente no lineales y no convexas. La resolución de este tipo de problemas se enfoca mediante el cálculo de la función de Gibbs, una función termodinámica de la temperatura, presión y composición que llega a un valor mínimo cuando el sistema se encuentra en equilibrio. Basados en este enfoque, los problemas de equilibrio de fases se pueden formular y resolver como problemas de optimización con la función de Gibbs como función objetivo (FO). Dependiendo del sistema y de sus condiciones de presión, temperatura y composición, la función de Gibbs puede presentar varios mínimos locales y un mínimo global que corresponde a la real condición de equilibrio [1]. La mayoría de los métodos tradicionales de optimización sólo encuentran soluciones locales, por lo que son fuertemente dependientes de los valores de inicialización del cómputo. El uso de técnicas de optimización global en estos

problemas está relativamente inexplorado y necesita mayor investigación [2]. Estas técnicas globales pueden clasificarse en deterministas y estocásticas. Si bien el primer tipo garantiza la convergencia a la solución global, los métodos deterministas actuales para problemas multivariados son muy costosos (en términos de esfuerzo computacional) y, en algunos casos es necesaria una reformulación del problema dependiendo de las características del modelo termodinámico en estudio. En cambio, los métodos de optimización global estocásticos pueden encontrar buenas soluciones en un tiempo razonable de cómputo, son fáciles de usar e independientes del modelo. Estos métodos pueden converger rápidamente al vecindario de la solución global y son adecuados para tratar un gran número de variables de decisión, ofreciendo un mejor compromiso entre la calidad de la solución y la eficiencia de cómputo. En el campo de la Termodinámica, los métodos de optimización estocásticos más usados son los Algoritmos Genéticos (GA) y la Búsqueda Tabú (TS). Entre otros problemas, estos métodos se aplicaron en cálculos de equilibrio y estabilidad de fase y en estimación de parámetros no lineales [3].

En este trabajo se presenta un algoritmo evolutivo con un operador genético sencillo basado en una búsqueda organizada a partir de la mejor solución. Su confiabilidad y eficiencia se evaluaron inicialmente mediante el cómputo de funciones multimodales propuestas como *benchmark*. Una vez validado, el algoritmo se aplicó a la solución de problemas de equilibrio de fases de sistemas termodinámicos. Se seleccionaron tres sistemas que de acuerdo a la literatura son los más adecuados para la evaluación de nuevos algoritmos de optimización, debido a que presentan mínimo local y global cercanos en el espacio de soluciones [4] [5] [6], situación que los hace desafiantes para el cómputo del equilibrio de fases. Para evaluar el desempeño del algoritmo propuesto en estos sistemas, los resultados se compararon con los de un algoritmo TS presentado en [6]. El resto de este artículo se estructura como sigue: en el apartado 2 se describe el modelo termodinámico y la FO, en el apartado 3 se muestra el algoritmo evolutivo propuesto y su proceso de validación, en el apartado 4 se describen los casos de estudio que se utilizaron, en el apartado 5 se muestra el estudio experimental y en el apartado 6 se presentan las conclusiones y se indican los trabajos futuros.

2 Modelo Termodinámico. Función Objetivo a Optimizar

Cada uno de los sistemas físicos se mantiene a temperatura y presión constantes y la finalidad es determinar la composición material del mismo que corresponde al equilibrio termodinámico a la temperatura y presión seleccionadas. Las cantidades de componentes en cada fase determinan su composición material (y por lo tanto su distribución) y son las variables de decisión del problema de optimización. Un sistema termodinámico queda definido por su temperatura, presión y componentes. Dado que los valores de composición que corresponden a la condición de equilibrio a la temperatura y presión fijadas, son los que minimizan la Función de Gibbs, se evalúan éstas como FO.

La función de Gibbs se evalúa mediante la ecuación (1).

$$G = \sum_{j=1}^{\pi} \sum_{i=1}^{nc} n_{ij} \mu_{ij} \quad (1)$$

Donde G es la función de Gibbs, π es el número de fases, nc es el número de componentes, n_{ij} y μ_{ij} son los números de moles y el potencial químico del componente i en la fase j , respectivamente.

Entonces, el problema de optimización global restringido queda como:

$$\min_n G$$

sujeto a las restricciones:

$$\sum_{j=1}^{\pi} n_{ij} = z_i F \quad 0 \leq n_{ij} \leq z_i F \quad (2)$$

Donde z_i es la fracción molar global para el componente i en la alimentación F .

La introducción de nuevas variables s_{ij} en lugar de las n_{ij} como variables de decisión elimina la restricción anterior y el problema de optimización se transforma en no restringido:

$$\begin{aligned} n_{i1} &= s_{i1} z_i F; i: 1 \dots nc \\ n_{ij} &= s_{ij} (z_i F - \sum_{m=1}^{j-1} n_{im}); i: 1 \dots nc; j: 1 \dots \pi - 1 \\ n_{i\pi} &= z_i F - \sum_{m=1}^{\pi-1} n_{im} \end{aligned} \quad (3)$$

Así, el problema de optimización se define como:

$$\min_n G \quad i: 1 \dots nc; j: 1 \dots \pi - 1; 0 \leq s_{ij} \leq 1$$

El espacio de soluciones queda restringido al entorno (0,1) para las variables s_{ij} ; éstas se transforman luego en cantidad de componentes mediante las ecuaciones (3).

Los valores de μ_{ij} se calculan mediante el empleo de ecuaciones de estado (EoS); las EoS empleadas en este trabajo son la de Peng-Robinson (PR-EoS) [7] y la de Soave-Redlich-Kwong. (SRK-EoS) [8].

3 Algoritmo Propuesto

3.1 Algoritmos Evolutivos

Los algoritmos evolutivos (AE) son metaheurísticas poblacionales que se vienen aplicando exitosamente en la solución de problemas complejos de varios dominios. Pertenecen a una clase de algoritmos de optimización que simulan la evolución de las especies y se basan en la idea de evolución de una población de individuos. Inicialmente esta población se genera de manera aleatoria donde cada individuo de la población se corresponde con una posible solución. Mediante la función objetivo se asocia a cada solución un valor de *fitness*, indicando su idoneidad para el problema. En cada paso, se selecciona el/los individuo/s que serán los padres, siguiendo el paradigma de selección en el que los individuos con mejor condición física tienen mayor probabilidad. Los individuos seleccionados se reproducen utilizando los operadores genéticos (cruce o mutación, entre otros) y generan nuevos descendientes. Finalmente, se aplica un esquema de sustitución para determinar qué individuos de la población (padres y descendientes) sobrevivirán. Este proceso se repite hasta que se cumpla un criterio de parada [9]. La Fig. 1 presenta el esquema general de un AE.

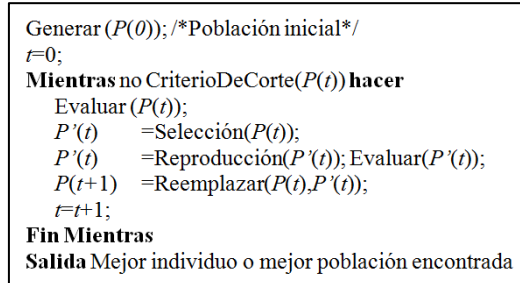


Fig. 1. Esquema general de un AE [9].

En los siguientes subapartados se presenta cómo el algoritmo propuesto se ajusta a dicho esquema, se describe su estructura y parámetros y luego se muestran los resultados obtenidos para los *benchmarks* que se utilizaron en su validación.

3.2 Estructura del algoritmo

En la fig. 2 se presenta la estructura básica del algoritmo propuesto en concordancia con el de la fig. 1. Más adelante se describe en detalle el operador genético implementado.

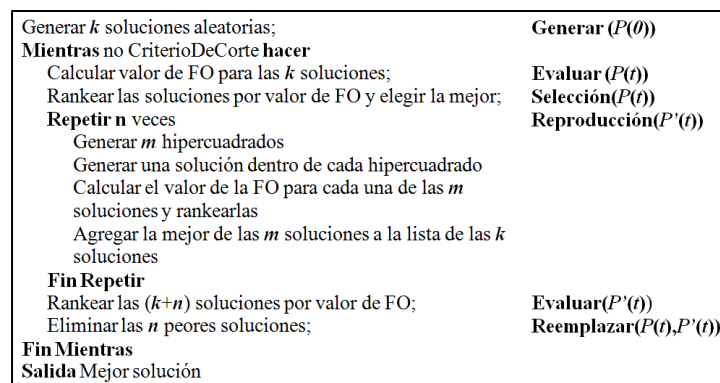


Fig. 2. Estructura del algoritmo de optimización propuesto.

Si bien los operadores genéticos básicos son mutación y cruce, algunas subclases de los AE como la Estrategia Evolutiva [10] sólo utilizan la mutación. Siguiendo esa línea, en este trabajo no se utiliza operador de cruce. Por lo tanto, el operador genético del presente algoritmo utiliza mutación para la generación de nuevas soluciones. Esto se hace, basado en [6], mediante la generación de nuevas soluciones a partir de la definición de hipercuadrados en el espacio de búsqueda. En cada iteración i se generan n nuevas soluciones a partir de la mejor solución obtenida en la iteración $i-1$. Para construir cada una de las soluciones se utilizan m hipercuadrados concéntricos, con centro en las coordenadas de la mejor solución usando el método de partición

geométrica, de acuerdo a la ecuación (4). La estructura de los hipercuadrados se muestra en la fig. 3.

$$sl_j = \frac{sl_m}{2^{m-j}} \text{ para } j = 1, 2, \dots, m \quad (4)$$

Donde sl_j es el semilado de cada uno de los m hipercuadrados y m , la cantidad de hipercuadrados. Una vez construidos los m hipercuadrados se genera aleatoriamente un punto dentro de cada uno de ellos (*zona 1*, *zona 2* y *zona 3* en la fig. 3). Se evalúa la función objetivo para cada uno de esos puntos y, finalmente, se agrega la mejor de las m soluciones a la lista de las k soluciones de la iteración $i-1$. El valor inicial del semilado sl_m , es un parámetro del algoritmo que se modifica en tiempo de ejecución en función de una variable que indica cuántas iteraciones han transcurrido sin obtener una mejor solución.

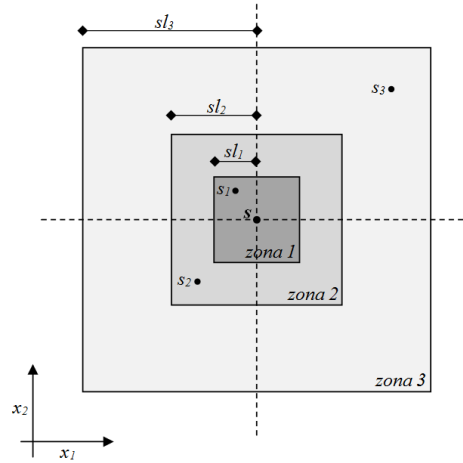


Fig. 3. Tres hipercuadrados en 2 dimensiones, cada uno conteniendo un punto generado aleatoriamente tomando como base el punto s .

3.3 Validación

Para evaluar la confiabilidad y eficiencia del algoritmo se seleccionaron de [11] y [12] cinco funciones multimodales para usarse como *benchmarks*. A continuación se presentan las ecuaciones de las funciones seleccionadas y en la tabla 1 se resumen sus detalles.

Función Easom (ES).

$$ES(x) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2) \quad (5)$$

Función Goldstein & Price (GP).

$$GP(x) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) * (30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2)) \quad (6)$$

Función Hartmann (H).

$$H_{3,4}(x) = -\sum_{i=1}^4 c_i \exp \left[-\sum_{j=1}^3 A_{ij} (x_j - P_{ij})^2 \right] \quad (7)$$

$$c = (1.0, 1.2, 3.0, 3.2)^T, A = \begin{bmatrix} 3.0 & 10.0 & 30.0 \\ 0.1 & 10.0 & 35.0 \\ 3.0 & 10.0 & 30.0 \\ 0.1 & 10.0 & 35.0 \end{bmatrix}, P = 10^{-4} \begin{bmatrix} 6890 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 0381 & 5743 & 8828 \end{bmatrix}$$

Función Shubert (SH).

$$SH = \left(\sum_{j=1}^5 j \cos((j+1)x_1 + j) \right) \left(\sum_{j=1}^5 j \cos((j+1)x_2 + j) \right) \quad (8)$$

Función Zhakarov (Z).

$$Z = \sum_{j=1}^n x_j^2 + \left(\sum_{j=1}^n 0.5jx_j \right)^2 + \left(\sum_{j=1}^n 0.5jx_j \right)^4 \quad (9)$$

Tabla 1. Detalle de las funciones usadas como *benchmark*

Función	Nº de Variables	Dominio de búsqueda	Mínimos locales	Mínimo global
ES	2	$-100 \leq x_i \leq 100$	Varios	$-1 \text{ en } x^* = (\pi, \pi)$
GP	2	$-2 \leq x_i \leq 2$	4	$3 \text{ en } x^* = (0,1)$
H	3	$0 \leq x_i \leq 1$	4	$-3.86278 \text{ en } x^* = (0.11461, 0.55564, 0.85254)$
SH	2	$-10 \leq x_i \leq 10$	760	18 mínimos globales SH=-186.7309
Z	N	$-5 \leq x_i \leq 10$	No tiene	$0 \text{ en } x^* = (0,0, \dots, 0)$

Para la ejecución de las funciones seleccionadas, el algoritmo se configuró con los siguientes parámetros:

- Número de soluciones candidatas. $k=10$
- Número de nuevas soluciones. $n=3$
- Número de hipercuadrados por solución. $m=3$
- Criterio de corte. $error (=solución \text{ deseada} - solución \text{ obtenida}) \leq 10^{-6}$
- Número máximo de iteraciones (si no llega al criterio de corte). $i=1000$
- Longitud inicial del semilado. $sl=0.25$

Se realizaron 100 ejecuciones para cada función, obteniéndose los resultados que se muestran en la Tabla 2. El número de éxitos hace referencia a los casos en que el algoritmo finalizó porque se cumplió el criterio de corte, mientras que el número de iteraciones y el tiempo de cómputo es el correspondiente al promedio de las soluciones exitosas en cada caso.

Tabla 2. Resultados obtenidos con las funciones usadas como *benchmark*

Función	Tasa de éxito (%)	Nº de iteraciones	Tiempo de cómputo (μseg)
ES	58	91	8724
GP	74	453	58716
H	69	870	207319
SH	79	419	108303
Z (n=2)	91	116	11868

4 Casos de Estudio

En la Tabla 3 se presentan los sistemas usados como casos de estudio de problemas de equilibrio de fases para evaluar la performance del algoritmo propuesto. Todos los sistemas presentan dos fases a la temperatura y presión mostradas en la misma tabla.

Tabla 3. Casos de Estudio (CE) propuestos.

CE	Componentes	T (K)	P (bar)	F	z_i	EoS
1	Metano + Sulfuro de Hidrógeno	190.0	40.53	1	0.9813; 0.0187	SRK
2	Dióxido de Carbono + Metano	220.0	60.80	1	0.20; 0.80	PR
3	Metano + Dióxido de Carbono + Sulfuro de Hidrógeno	227.6	48.60	1	0.4989; 0.0988; 0.4023	PR

Teh y Rangaiah [6] evaluaron la performance de un algoritmo TS aplicado a estos sistemas, informando los valores de la FO obtenidos y definiendo los siguientes indicadores: número promedio de evaluaciones de la FO y tiempo promedio de cómputo. Para este trabajo, se usó el primero como indicador de comparación dado que, debido a la diferencia de hardware, el último no es comparable.

5 Estudio Experimental

Para cada uno de los casos de estudio propuestos, se empleó el algoritmo evolutivo con operador genético sencillo con el propósito de minimizar la función de Gibbs. Para cada CE se realizaron 100 ejecuciones con diferentes semillas de números aleatorios. Para cada ejecución, se registró valor de la FO, tiempo de ejecución y número de evaluaciones de la FO (iteraciones). La Tabla 4 resume los resultados obtenidos en la evaluación de cada CE.

Tabla 4. Resultados obtenidos en los CE

CE	Tasa de éxito (%)	Nº de iteraciones	Tiempo de cómputo (μ seg)
1	38	118	760421
2	57	61	392667
3	76	678	4228960

5.1 Indicadores de Performance

En este apartado se evalúa la performance del algoritmo propuesto contrastando los resultados de 25 ejecuciones¹ para cada CE con idéntico número de ejecuciones del algoritmo TS implementado en [6].

Los resultados para los valores obtenidos de la FO se presentan en la Tabla 5. El

¹ Se toman sólo 25 ejecuciones a efectos comparativos, dado que Teh & Rangaiah [6] informan los valores de performance para esta cantidad de ejecuciones

error relativo de cómputo entre ambos algoritmos se considera no significativo.

Tabla 5. Comparación de valores FO obtenidos.

EC	Valor de FO obtenido por el algoritmo propuesto	Valor de FO obtenido por el algoritmo de [6]	Error relativo (%)
1	-0.454022	-0.453972	0.01
2	-0.954313	-0.951211	0.33
3	-1.863369	-1.859754	0.19

En la tabla 6 se presentan los resultados de la comparación entre el número promedio de evaluaciones de la FO. Se aprecia que este número es sensiblemente menor para el algoritmo propuesto.

Tabla 6. Comparación del número promedio de evaluaciones de la FO

CE	Nº promedio de evaluaciones de la FO en este trabajo	Nº promedio de evaluaciones de la FO en [6]
1	630	1268
2	69	1424
3	1319	3505

La ecuación (10) permite calcular la diferencia relativa entre el valor de la función de Gibbs tomada en una iteración cualquiera respecto a su valor final. Esta diferencia se usa en las Fig. 4, 5 y 6 para presentar la evolución entre el mejor valor calculado de la función objetivo en cada una de las iteraciones (G_{itera}) y el valor óptimo obtenido (G_{min}).

$$\%D_{relat} = \left| \frac{G_{itera} - G_{min}}{G_{min}} \right| 100 \quad (10)$$

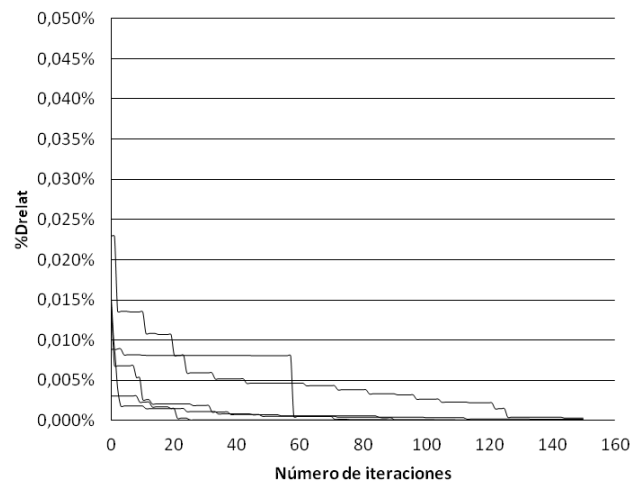


Fig. 4. Diferencia relativa vs. Nº de iteraciones para el CE 1

Según se muestra en dichas figuras, la diferencia relativa es siempre inferior a 0.025 %, a excepción de una ejecución en el CE 3, cumplidas 20 iteraciones, independientemente de las semillas que se utilicen para generar las soluciones. Ello podría ser una indicación de la robustez del algoritmo implementado.

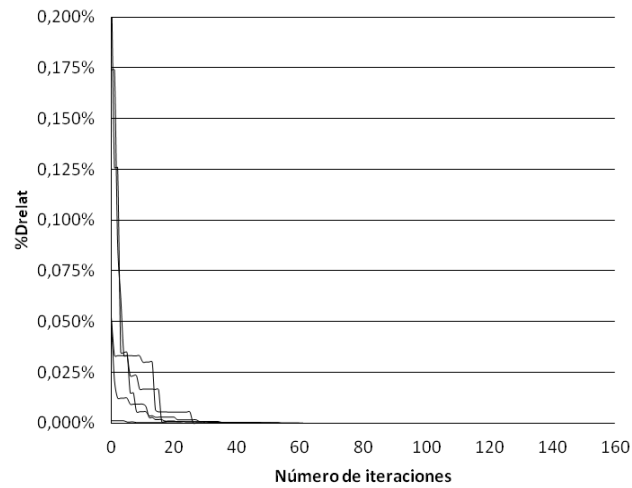


Fig. 5. Diferencia relativa vs. N° de iteraciones para el CE 2

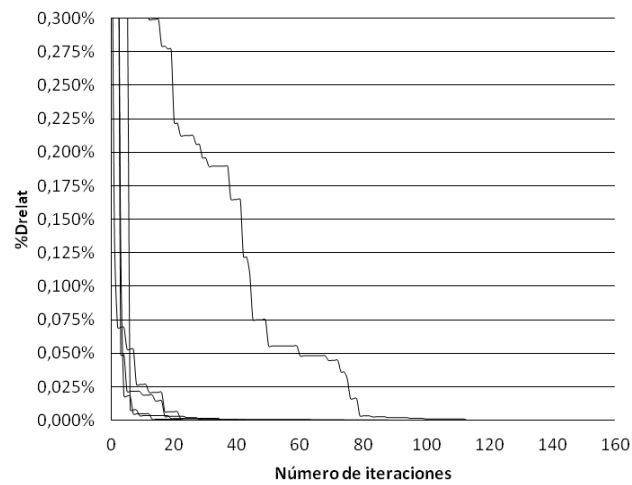


Fig. 6. Diferencia relativa vs. N° de iteraciones para el CE 3

6 Conclusiones y Trabajos Futuros

En este trabajo se presentó un algoritmo evolutivo con un operador genético sencillo

cuya confiabilidad y eficiencia se evaluaron mediante el cómputo de funciones multimodales. Una vez validado, el algoritmo se aplicó a la solución de problemas de equilibrio de fases de sistemas termodinámicos.

El algoritmo propuesto se comporta de manera adecuada en los cálculos de mínimos de las funciones multimodales seleccionadas como *benchmarks*.

De igual modo, y en comparación con otro algoritmo aplicado a la minimización de la función de Gibbs para el cómputo de equilibrio termodinámico de fases, los resultados obtenidos se consideran satisfactorios, al menos en los CE presentados.

Como trabajos futuros se pretende, por un lado verificar la robustez del algoritmo siempre en este campo pero evaluando su performance de sistemas más complejos; y por otro lado, con el propósito de mejorar la tasa de éxitos, se propone incluir un operador de cruce, determinar cómo influye en la performance del algoritmo modificaciones en sus parámetros y evaluar otros operadores genéticos que permitan realizar búsqueda global, teniendo en cuenta que el mínimo global de la función de Gibbs puede encontrarse muy cerca a mínimos locales, y/o diferir muy poco entre sí en sus valores.

Referencias

1. L. E. Baker, A. C. Pierce, K. D. Luks "Gibbs Energy Analysis of Phase Equilibria" Society of Petroleum Engineers Journal, pp 731-742, 1982.
2. G.P. Rangaiah "Evaluation of genetic algorithms and simulated annealing for phase equilibrium and stability problems", Fluid Phase Equilibria Vol 187-188pp. 83-109 2001.
3. H. Zhang, A. Bonilla-Petriciolet, and G. P. Rangaiah, "A Review on Global Optimization Methods for Phase Equilibrium Modeling and Calculations," *The Open Thermodynamics Journal*. Vol. 5, (Suppl 1-M7) pp. 71–92, 2011.
4. Y. Sofyan, A.J. Ghajar and K.A. M. Gasem. "Multiphase Equilibrium Calculations using Gibbs minimization techniques" *Ind. Eng. Chem. Res.* Vol. 42 pp 3786-3801, 2003
5. A. C. Sun, W. D. Seider. "Homotopy- continuation method for stability analysis in the global minimization of the Gibbs free energy" *Fluid Phase Equilibria*, Vol. 103 pp. 213-249, 1995
6. Y. S. Teh, G.P. Rangaiah. "Tabu Search for global optimization of continuous functions with application to phase equilibrium calculations" *Computers and Chemical Engineering*. Vol 27, pp 1665-1679, 2003-
7. D.Y. Peng, D.B. Robinson "A new two constants equation of state" *Industrial Engineering Chemical Fundamental*. Vol 15 pp 59-63, 1976.
8. G. Soave "Equilibrium constants from a modified Redlich-Kwong equation of state" *Chemical Engineering Science* Vol. 27 pp.1197-1203, 1972.
9. E. G. Talbi, *From design to implementation*. Wiley, 2009.
10. I. Rechenberg. Cybernetic solution path of an experimental problem. Technical Report, Royal Aircraft Establishment Library Translation No. 1112, Farnborough, UK, 1965.
11. http://www.geatbx.com/docu/fcnindex-01.html#P86_3059 (Fecha de última visita: 18-jun-2012)
12. http://www-optima.amp.i.kyotou.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm (Fecha de última visita: 18-jun-2012)