

Un modelo de optimización estocástica aplicado a la optimización de estructuras de barras prismáticas

Salvador Botello

Universidad de Guanajuato
Facultad de Ingeniería Civil
Avenida Juárez 77, 36000 Guanajuato, México
Tel.: 52-473-2 07 79, Fax: 52 473 2 32 41
e-mail: botello@quijote.ugto.mx

José Luis Marroquín

Centro de Investigación en Matemáticas
Apdo. Postal 402, 36000 Guanajuato, México
Tel.: 52-473-2 71 55, Fax: 52-473-2 57 49
e-mail: jlm@fractal.cimat.mx

Eugenio Oñate

ETS de Ingenieros de Caminos, Canales y Puertos
Universidad Politécnica de Cataluña
Gran Capitán s/n, Módulo C1
08034 Barcelona, España
Tel.: 34-93-205 70 16, Fax: 34-93-401 65 17
e-mail: onate@cimne.upc.es

Resumen

Se presenta de forma general una familia de algoritmos de búsqueda estocástica en forma parametrizada que puede ser utilizada para resolver problemas de optimización combinatoria. Esta familia incluye los ya conocidos algoritmos de optimización: algoritmos genéticos, estrategias evolutivas y recocido simulado. Además se incluye una nueva técnica propuesta por los autores del trabajo que combina el recocido simulado en paralelo con selección. Se muestra el buen comportamiento del algoritmo propuesto y se aplica en la optimización de una armadura para una nave industrial en acero sujeta a cargas laterales, utilizando como información un catálogo de secciones transversales. También se estudia en forma óptima la estructura de un puente considerando efectos de compresión en el diseño de los elementos estructurales. Los autores del trabajo están aplicando esta técnica para realizar optimización de un gran número de problemas de ingeniería civil y de mecánica.

A MODEL FOR STOCHASTIC OPTIMIZATION APPLY OF PRISMATIC BAR STRUCTURES

Summary

In this paper we present a general parametric family of stochastic search algorithms for solving combinatorial optimization problems. This family includes well known algorithms such as Genetic Algorithms, Evolutionary Strategies and Simulated Annealing as particular instances, and also a new scheme that combines parallel Simulated Annealing with selection. A convergence proof and analytic results that approximate the invariant distribution of states for this new algorithm are given. Also, we show experimentally that by an appropriate choice of parameters, one can find an algorithm in the class that exhibits significantly better experimental performance than the known methods in a variety of situations; these include: deceptive problems, cooperation and games and structural design in civil engineering applications.

INTRODUCCIÓN

Las técnicas de búsqueda estocástica se han aplicado para resolver una gran variedad de problemas complejos de optimización combinatoria. La más antigua de ellas es probablemente recocido simulado (Simulated Annealing) (RS)⁹ que genera una secuencia de soluciones por combinación de operaciones de mutación con un criterio de aceptación que es más estricto al incrementar el tiempo². Otras técnicas como las estrategias evolutivas (EE)^{13,5} y los algoritmos genéticos (AG)⁸ también involucran una operación de mutación pero ésta es aplicada a todos los elementos de la población, los cuales compiten unos con otros (en la selección) e intercambian información (como el caso de cruzamiento en los AG).

Dichas técnicas han sufrido gran variedad de combinaciones y modificaciones^{1,3,10,16}. Una de las más importantes es probablemente la de recocido simulado paralelo recombinado (RSPR)¹² que utiliza una población inicial generada en forma aleatoria, *cross-over* (cruce de familias) y mutaciones en cada paso de tiempo. En este artículo se generaliza esta idea y se introduce la selección en cada paso, lo que permite realizar una mejora en una variedad de situaciones. Como se verá más adelante, RS, EE, AG y RSPR son en realidad situaciones particulares de una familia de algoritmos con cuatro parámetros, y por consiguiente es posible encontrar unos valores específicos para los parámetros que mejoran la solución, con respecto a cada una de las técnicas conocidas.

En la sección siguiente se indican los operadores básicos y además se presenta la forma paramétrica general de la familia de algoritmos de búsqueda estocástica en paralelo. Para mostrar las bondades de esta nueva técnica se presentan ejemplos de aplicación.

UNA FAMILIA DE ALGORITMOS DE BÚSQUEDA ESTOCÁSTICA

El problema general que nosotros tratamos de resolver es el siguiente: teniendo un *espacio de estado* $\Omega = Q_1 \times \dots \times Q_n$ donde $Q_i, i = 1, \dots, n$ son conjuntos de tamaño finito y una función $U : \Omega \mapsto \mathbf{R}$ que puede ser llamada *función de costo*, deseamos encontrar un vector $x^* = (x^{*1}, \dots, x^{*n}) \in \Omega$ que minimice U globalmente.

Los componentes básicos del algoritmo son una *población* X , formada por una serie de individuos (x^1, \dots, x^N) , y un conjunto de operadores que se aplican iterativamente para generar nuevos operadores, llamados: selección, cruzamiento (*cross-over*), mutación y aceptación. Cada uno de ellos está caracterizado por un parámetro que representa una elección particular entre la exploración y la explotación del espacio de búsqueda. A continuación se realiza una breve descripción de estos operadores.

Operadores

Selección

Para una función que mide la bondad de una solución de forma parametrizada $f_\gamma : \Omega \mapsto \mathbf{R}$, la selección consiste en encontrar los N elementos de una población determinada, con una probabilidad proporcional al valor de dicha función. Con objeto de reducir la variancia, sopesando la similaridad y diversidad, algunos procedimientos especiales han sido propuestos como por ejemplo la *técnica estocástica de residuos* y la *selección universal estocástica*¹¹, eventualmente combinadas con algunas adaptaciones específicas para cada uno de los algoritmos de búsqueda estocástica, por ejemplo *selección justa*⁷, introducción de especies en la población o la adaptación de la función de costo.

Debido a su simplicidad y eficiencia, se seleccionó la *técnica estocástica de residuos*, cuyo objetivo principal es colocar de forma determinística en la nueva población un número

SCIPEDIA

Register for free at <https://www.scipedia.com> to download the version without the watermark

Operador $S_\gamma(X)$	
Calcula	para cada elemento x_k de X , la aceptación relativa f_k truncando el número de ocurrencias esperado, será n_k : $f_k = \frac{f_\gamma(x_k)}{\sum_{j=1}^N f_\gamma(x_j)}$ $n_k = \text{Int}(Nf_k)$
Donde	la función $\text{Int}()$ calcula la parte entera del argumento
Hace	$t = \sum_{k=1}^N n_k$ y construye el bloque (y_1, \dots, y_t) haciendo n_k copias de cada elemento x_k ;
Elige	y_{t+1}, \dots, y_N de X con probabilidad de seleccionar el elemento x_k igual a $\frac{Nf_k - n_k}{\sum_{j=1}^N (Nf_j - n_j)}$
Hacer	$S_\gamma(X) = (y_1, \dots, y_N)$

Algoritmo 1

esperado de copias de cada elemento seleccionado (truncando a un número natural) y colocando las posiciones remanentes en la forma clásica (estocástica).

El resultado es una familia de operadores de selección de un parámetro $S_\gamma : \Omega^N \mapsto \Omega^N$ la cual está definida en el Algoritmo 1. La familia de funciones que mide la bondad f_γ puede variar entre $\gamma = 0$, y $\gamma = 1$. Por ejemplo, si para todo $x \in \Omega$, la función de costo $U(x)$ es siempre positiva, tendremos

Register for free at <https://www.scipedia.com> to download the version without the watermark

$$f_\gamma(x) = 1 - \gamma(1 + U(x)) \quad (1)$$

con $\gamma \in [0, 1]$.

Es posible también utilizar la forma exponencial de la forma⁴

$$f_\gamma(x) = \exp[-\gamma U(x)] \quad (2)$$

donde para valores grandes de γ se obtiene un operador S_γ el cual encuentra el mejor elemento de una población y lo reproduce N veces.

Mutación

Podemos definir una familia de operadores de mutación con un parámetro $M_\mu : \Omega^N \mapsto \Omega^N$ por medio del Algoritmo 2:

La probabilidad de mutación $p(\mu, x, X)$ puede ser uniforme (i.e, $p(\mu, x, X) = \mu$) o adaptativa según el valor de aceptación de x . Podemos escribir la probabilidad adaptativa como¹⁴

$$p(\mu, x, X) = \begin{cases} \mu \frac{f_{\max} - f(x)}{f_{\max} - \bar{f}} & \text{if } f(x) > \bar{f} \\ \mu & \text{if } f(x) \leq \bar{f} \end{cases} \quad (3)$$

Operador $M_\mu(X)$	
Para cada elemento x de la población hacer:	
Construir un elemento y como sigue:	
Para cada componente x^i de x hacer:	
$y^i = r^i$ con prob. $p(\mu, x, X)$	
$= x^i$ con prob. $1 - p(\mu, x, X)$	
donde r^i es el elemento seleccionado de forma aleatoria Q_i con una probabilidad uniforme	
Hacer	$M_\mu(X) = (y_1, \dots, y_N)$

Algoritmo 2

donde f_{\max} y \bar{f} son el valor máximo y promedio de la función de aceptación de la población respectivamente. Es esencial para el mejor funcionamiento de este operador que el mejor individuo de la población sea transmitido por el operador sin mutarlo. Para realizarlo, se define $p(\mu, x, X) = 0$ si x es el mejor elemento en X e igual a μ para los otros casos.

Cruzamiento

Este operador realiza el intercambio de información mediante la división de dos elementos de la población en dos partes y concatenándolas de forma cruzada en cada uno de ellos.

Operador $C_\xi(X)$	
Si	bien algunos elementos de X no se han utilizado en <i>cross-over</i> :
	Selecciona 2 elementos no utilizados x_j, x_k de X con probabilidad $p(\xi, X, x_j, x_k)$, hacer cruzamiento:
	Determinar una posición r entre 1 y n para una distribución uniforme;
	Construir $y_j = (x_j^1, \dots, x_j^r, x_k^{r+1}, \dots, x_k^n)$
	$y_k = (x_k^1, \dots, x_k^r, x_j^{r+1}, \dots, x_j^n)$
Si no	Colocar $y_j = x_j, y_k = x_k$
Hacer	$C_\xi(X) = (y_1, \dots, y_N)$

Algoritmo 3

Con el fin de imponer diversidad se utilizan todos los elementos a la vez en el operador de cruzamiento y se debe garantizar que no se pierda el mejor de todos ellos.

El resultado es una familia de operadores de cruzamiento de un parámetro $C_\xi : \Omega^N \mapsto \Omega^N$ definida en el Algoritmo 3, donde la probabilidad de cruzamiento $p(\xi, X, x, y)$ puede ser uniforme o variar en el tiempo dependiendo del valor de aceptación del elemento implicado.

Sin embargo, se impondrá $p(\xi, X, x, y) = 0$ si x o y es el mejor elemento de la población y hará igual a ξ , en los otros casos.

Aceptación

Finalmente, para realizar la selección entre una población X y un candidato de la población mutada Y se aplica el criterio de aceptación *metrópolis* a cada elemento de X y Y . Esto define una familia de operadores de aceptación $A_\beta : \Omega^N \times \Omega^N \mapsto \Omega^N$, la cual es descrita en el Algoritmo 4.

Operador $A_\beta(X, Y)$	
Para todos los elementos $x_k \in X, y_k \in Y$ hacer:	
Hacer	$\Delta U = U(y_k) - U(x_k)$
Si	$\Delta U \leq 0$, hacer $u_k = y_k$;
Si	$\Delta U > 0$, hacer $u_k = y_k$ con prob. $\frac{\exp[-\beta\Delta U]}{1 + \exp[-\beta\Delta U]}$ $u_k = x_k$ con prob. $\frac{1}{1 + \exp[-\beta\Delta U]}$
Hacer	$A_\beta(X, Y) = (u_1, \dots, u_N)$

Algoritmo 4

Algoritmo general de búsqueda estocástica

Definición

El algoritmo general de búsqueda estocástica (AGBE) se define por el sistema dinámico

$$X^{(t+1)} = A_\beta(Y^{(t)}, M_\mu(C_\xi(Y^{(t)}))) \quad (4)$$

Register for free at <https://www.scipedia.com> to download the version without the watermark

$$Y^{(t)} = S_\gamma(X^{(t)})$$

donde en general los parámetros β, μ, ξ y γ pueden ser variables en el tiempo y la población inicial $X^{(0)}$ es determinada de forma aleatoria.

Se pueden determinar de forma simple los valores específicos de los parámetros para los siguientes casos:

- para recocido simulado (RS) clásico con N puntos de inicio: $\xi = \gamma = 0$; β incrementándose;
- estrategia evolutiva: $\beta = \xi = 0$;
- algoritmo genético: $\beta = 0$;
- recocido simulado paralelo recombinado : $\gamma = 0$; β incrementándose.

Se pueden encontrar valores apropiados diferentes de cero para todos los parámetros para incrementar la eficacia de los algoritmos. Las pruebas de convergencia pueden verse en la referencia ¹⁵.

APLICACIÓN A LA OPTIMIZACIÓN DE UNA ESTRUCTURA PLANA DE BARRAS UNIDA CON RÓTULAS

Estudiaremos el comportamiento del AGBE para un problema “académico”: la optimización de estructuras planas con rótulas⁶ que puede ser utilizada para soportar el techo de naves industriales.

La optimización de la estructura consiste en encontrar la sección transversal de cada elemento (de un conjunto discreto, por ejemplo, de un catálogo), de tal forma que todos las tensiones que se encuentren sobre ella sean menores que un valor permisible, y el peso total de la estructura sea lo menor posible.

Llamando $x(e)$ la entrada de catálogo seleccionada para cada elemento e de la estructura, la función de costo a ser minimizada será

$$U(x) = \sum_e [\rho_{x(e)} A_{x(e)} L_e + \lambda(\delta\sigma_e)]$$

donde ρ_x , A_x y L_e son el peso específico, la sección transversal de catálogo $x(e)$ y la longitud del elemento e ; $\delta\sigma_e$ es la cantidad de tensión que excede el valor máximo permisible y λ un factor de penalización para la tensión en exceso (en nuestro caso consideramos $\lambda = 10000$).

En la Figura 1 se muestra la estructura con rótulas, la cual se encuentra sujeta a 11 cargas horizontales de 4,994 kg cada una de ellas y con condiciones de apoyo en su base. El material utilizado para todas las barras es acero con un módulo elástico de $2,1 \times 10^6$ kg/cm² y la tensión máxima permisible es de 3,500 kg/cm². Las secciones transversales posibles van de 0,1 hasta 50,1 cm² con incrementos de 0,5 cm², de tal forma que el espacio de estados para cada elemento es de tamaño 100. Considerando de antemano que la estructura debe de ser simétrica, se determinan solamente 25 tipos de materiales en las 49 barras que forman la estructura.

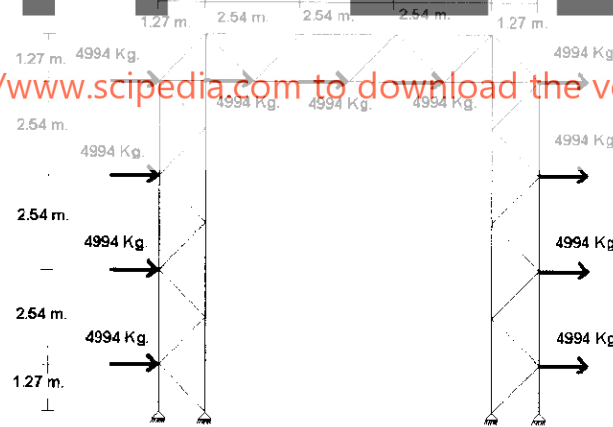
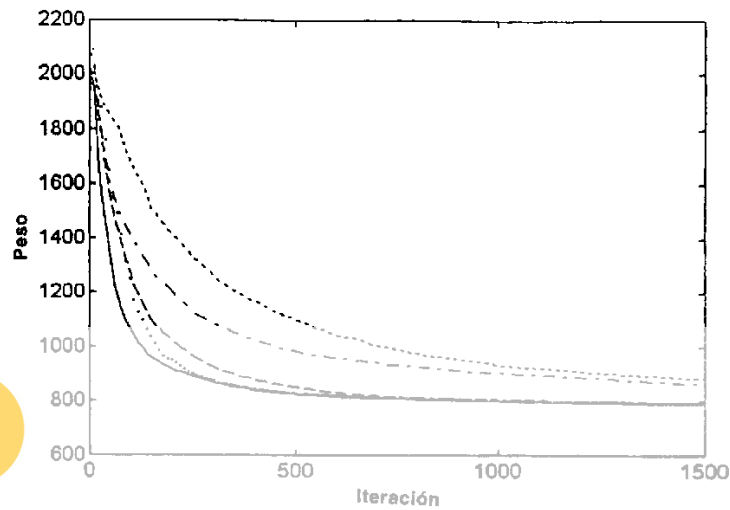


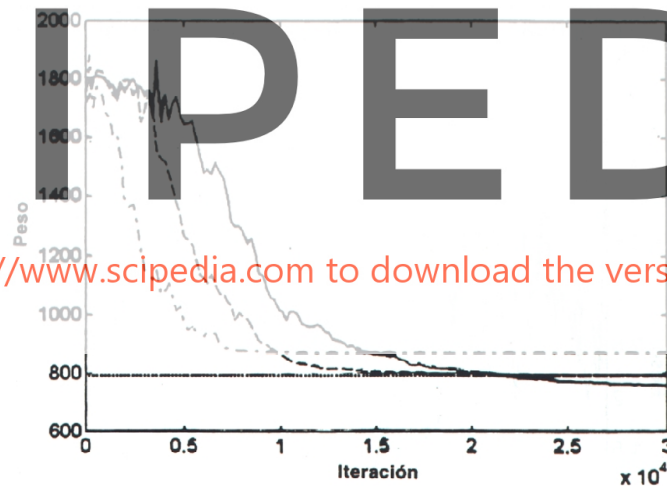
Figura 1. Estructura de barras unidas con rótulas de una nave industrial utilizada para encontrar su peso mínimo cumpliendo condiciones de servicio

Consideramos una población de tamaño $N = 20$ y optimizamos la estructura para diferentes parámetros del AGBE. Los resultados se muestran en la Figura 2. Podemos ver en (a) los resultados obtenidos con los parámetros $\beta = 1,01$, $\mu = 0,1$, $\gamma = 1$ (con escalado lineal) y $\xi = 0,8$ que ofrecen la mejor relación entre velocidad de convergencia y calidad de la solución. En una máquina con 20 procesadores podríamos obtener resultados 20 veces más rápidos que los que se obtienen utilizando un algoritmo de RS. En (b) se muestran las diferentes trayectorias para RS con diferentes temperaturas. Notar que en una máquina

secuencial. El comportamiento óptimo (en términos de la carga computacional total y la calidad de la solución) se obtiene precisamente en este caso ¹⁵.



(a)



(b)

Figura 2. Resultados de la optimización en peso de la estructura de la nave industrial de la Figura 1. (a) Comportamiento de la convergencia (sobre 20 corridas MonteCarlo) de: estrategia evolutiva (línea discontinua delgada), algoritmos genéticos (línea discontinua gruesa), AGBE para $\xi = 0$ sin *cross-over* (línea discontinua), RSPR (línea punteada) y AGBE con *cross-over* (línea sólida). (b) Comportamiento del RS convencional para el problema de la nave industrial para $\beta = 1,001$ (curva punteada gruesa), $\beta = 1,0002$ (curva punteada) y $\beta = 1,0001$ (curva sólida). La línea horizontal indica el peso obtenido para AGBE después de 1500 iteraciones (Notar que las escalas en (a) y (b) son diferentes)



Register for free at <https://www.scipedia.com> to download the version without the watermark

OPTIMIZACIÓN DE LA ESTRUCTURA DE UN PUENTE FORMADO POR BARRAS UNIDAS CON RÓTULAS

Como otro ejemplo se presenta la optimización de un puente formado de 213 barras con un máximo de 96 tipos de barras diferentes (por condiciones de simetría), con una longitud de 24 metros de claro (formado por 12 celdas de 2 metros cada una), con una altura máxima de la estructura de 8 metros, sujeta a una carga carga uniforme de 800 kilogramos por cada celda y una carga lateral que tiene un valor máximo en su extremo superior de 1000 kilogramos (Figura 3). El catálogo utilizado es de la empresa AHMSA (Altos Hornos de México, S.A.), en el que se incluyen 233 perfiles estructurales.

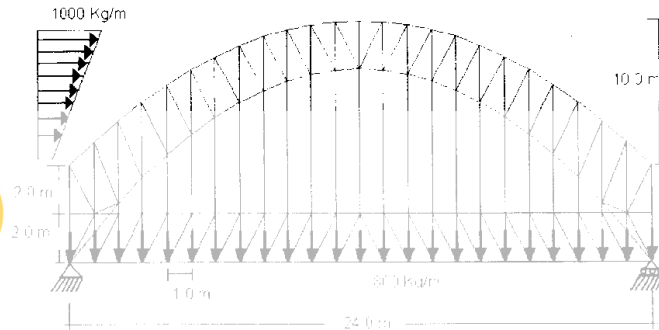


Figura 3. Estructura de barras unidas con rótulas para un puente de 24 metros de claro para ser optimizado con un catálogo de secciones real

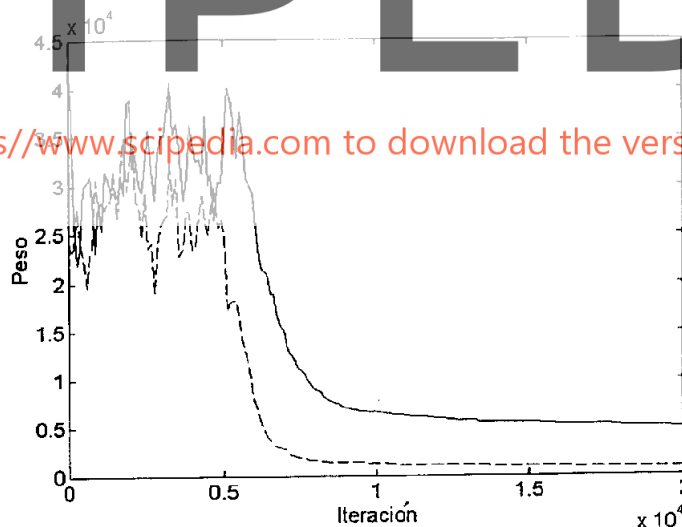


Figura 4. Resultados de la optimización en peso de la estructura del puente de la Figura 3. Comportamiento de la convergencia para recocido simulado RS utilizando reducción de la tensión permisible en diseño de elementos (línea continua) y sin utilizar dicha reducción (línea discontinua)

El acero utilizado tiene una fluencia de 2500 kg/cm^2 . En dicho catálogo encontramos el área y los momentos de inercia en ejes principales de cada uno de los perfiles. Para realizar el diseño de los elementos se utilizaron las especificaciones del AISC (American Institute of Steel Construction) bajo concepto de tensiones de servicio. La función de costo

utilizada es similar a la descrita en (10). El tamaño de la población utilizada es de 1 (dado que trabajamos en una máquina con un sólo procesador), con un factor de $\beta = 1,001$, realizando 20000 iteraciones. El resultado del comportamiento de la optimización puede verse en la Figura 4. En la referencia¹⁵ se muestran las secciones finales recomendadas, indicándose además un porcentaje de eficiencia de tensiones de trabajo contra resistentes en cada una de ellas. Cabe destacar que la estructura tiene un peso menor cuando no se considera el efecto de reducción de resistencia para tensiones de compresión, como es lógico esperar.

CONCLUSIONES

Se presenta una familia de búsqueda estocástica en paralelo de cuatro parámetros que incluye como casos particulares varias técnicas populares, como son los algoritmos genéticos, estrategias evolutivas y recocido simulado (con múltiples puntos de inicio) y algunas técnicas híbridas como el recocido simulado paralelo recombinado. También se incluye un nuevo algoritmo que combina el recocido simulado en paralelo con selección.

El hecho de tener una familia paramétrica de algoritmos abre la posibilidad de utilizar técnicas de búsqueda numérica para encontrar los parámetros óptimos para cada clase de problemas en particular. Experimentalmente, pudimos determinar que estos parámetros óptimos se encuentran en general en el interior de un espacio de parámetros.

Los mejores resultados se obtuvieron para técnicas de recocido simulado en paralelo con interacción ya sea con la selección o con cruzamiento, ambos con aproximadamente el mismo tipo de respuesta. Para el caso en que los elementos de una cadena de cromosomas no tengan interacciones localizadas, el efecto del cruzamiento es menor que el de selección¹⁵, motivo por el cual la forma más general de obtener una buena solución consiste en utilizar el AGBE combinando la selección con el cruzamiento, obteniendo de esta forma en general, los resultados mejores y más robustos.

La rapidez de la convergencia del algoritmo implica que el tamaño de la población se incrementa; sin embargo la carga computacional se aumenta. El mejor de los resultados puede ser obtenido utilizando máquinas paralelas donde el número de procesadores sea igual al número de iteraciones de tal forma que el costo computacional sea el mismo que el obtenido en particular para máquinas en serie, los mejores resultados se obtienen utilizando el algoritmo común de recocido simulado, exceptuando los casos en que se prefiere obtener una solución razonable (alrededor de un subóptimo) en el menor tiempo posible. En este caso el AGBE con una población de pequeño tamaño (por ejemplo $N = 4$) se pueden obtener buenos resultados y una muy buena relación entre calidad y costo computacional, trabajando en máquinas de un sólo procesador.

Finalmente, se puede considerar extender el algoritmo con más operadores y parámetros. Un candidato es incluir la diversidad de forma explícita. Sin embargo en algunos casos puede ser posible obtener mejores resultados, si exceptuamos el caso en que se desean obtener todos los óptimos¹¹. La ganancia obtenida por este operador adicional es menor que el costo de encontrar los valores óptimos del nuevo parámetro.

REFERENCIAS

- 1 D.H. Ackley, “*A connectionist machine for genetic hillclimbing*”, Kluwer Academic Publishers, Boston, (1987).
- 2 S. Anily y A. Federgruen, “Simulated annealing methods with general acceptance probabilities”, *J. Applied Prob.*, Vol.24, pp. 657–667, (1987).

- 3 T. Boseniuk y W. Ebling, “Boltzmann-, Darwin- and Haeckel–strategies in optimization problems”, *Lecture Notes in Computer Science: Parallel Problem Solving from Nature*, Vol. **496**, pp. 430–444, (1991).
- 4 M. de la Maza y B. Tidor, “Boltzmann weighted selection improves performance of genetic algorithms”, A.I. Memo 1345, Artificial Intelligence Lab., Massachusetts Institute of Technology, Cambridge, MA, (1991).
- 5 L.J. Fogel, A.J. Owens y M.J. Walsh, “*Artificial intelligence through simulated evolution*”, New York, Wiley Pub., (1966).
- 6 M. Galante, “Genetic algorithms as an approach to optimize real–world trusses, *International Journal for Numerical Methods in Engineering*, Vol. **39**, pp. 361–382, (1996).
- 7 D.E. Goldberg, “A note on Boltzmann tournament selection for genetic algorithms and population–oriented simulated annealing”, *Complex Systems*, Vol. **4**, pp. 445–460, (1990).
- 8 D.E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison Wesley, Reading, MA, (1989).
- 9 S. Kirkpatrick, C.D. Gelatt y M.P. Vecchi, “Optimization by simulated annealing” *Science*, Vol. **220**, 4598, pp. 671–680, (1983).
- 10 F.T. Lin, C.Y. Kao y C.C. Hsu, “Incorporating genetic algorithms into simulated annealing” *Proc. of the Fourth Int. Symposium on Artificial Intelligence*, pp. 290–297, (1991).
- 11 S.W. Mahfoud, “Niching methods for genetic algorithms”, Doctoral dissertation, University of Illinois, (1995).
- 12 S.W. Mahfoud y D.E. Goldberg, “Parallel recombinative simulated annealing: a genetic algorithm”, Tech. Report, Depr. of Computer Science, Univ. of Illinois, (1994).
- 13 I. Rechenberg, “*Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*”, Fromman, Stuttgart, (1973).
- 14 M. Srinivas y L.L. Patnaik, “Adaptive probabilities of crossover and mutation in genetic algorithms”, *IEEE Trans. on Syst., Man, and Cyber.* Vol. **24**, 5, pp. 656–667, (1994).
- 15 J.L. Marroquín, S. Botello y J.V. Horebeek, “A family of parallel stochastic search algorithms”, *Comunicaciones del Cimat*, pp. 183, (1996).
- 16 J. Estupiñán, E. Oñate y B. Suárez, “Optimización topológica mediante algoritmos genéticos, estrategias evolutivas y el método de Baluja”, *Revista Int. Métodos Numéricos para Cálculo y Diseño Ingeniería*, Vol. **14**, N° 4, pp. 427–438, (1998).

Register for free at <https://www.scipedia.com> to download the version without the watermark