

**UNA FAMILIA DE ALGORITMOS ESTOCASTICOS EN
PARALELO APLICADOS A LA OPTIMIZACION
DE ESTRUCTURAS DE ACERO**

Salvador Botello

Universidad de Guanajuato
Facultad de Ingeniería Civil
Av. Juárez 77, Guanajuato, Gto. 36000, México.

José L. Marroquín, Johan Van Horebeek

Centro de Investigación en Matemáticas A. C.
Apdo. Postal 402, Guanajuato, Gto. 36000, México.

RESUMEN

Se presenta de forma general una familia de algoritmos de búsqueda estocástica en forma parametrizada 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 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. Los autores del trabajo están aplicando ésta técnica para realizar optimización de un gran número de problemas de Ingeniería Civil y de Mecánica.

ABSTRACT

A general parametric family of stochastic search algorithms for solving combinatorial optimization problems it is presented. This family includes well known algorithms such as Genetic Algorithms, Evolutionary Strategies and Simulated Annealing. Moreover it is include a new scheme that combines parallel Simulated Annealing with selection. This work is presented to show the apod behaviour of the proposed algorithm and it is applied on optimization of steel structure under lateral loads, using information base on steelers catalog for transversal section. Authors are applying this technique to optimize different problems on Civil and Mechanical Engineering.

1. Introducción

Se han utilizado varios tipos de técnicas de búsqueda estocástica aplicados a resolver una gran variedad de problemas complejos de optimización combinatoria. El más antiguo de ellos 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 se hace más estricto al incrementar el tiempo. 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 cruces por medio de cross-over en los AG).

Dichas técnicas han sufrido gran variedad de combinaciones y modificaciones [1], [3], [10]. 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 muestra la inclusión de la selección en cada paso, lo que permite realizar una mejora en una gran variedad de situaciones. Como se verá más adelante, RS, EE, AG y RSPR son en realidad situaciones particulares de una familia de algoritmos con 4 parámetros, y por consiguiente es posible encontrar valores específicos para los parámetros que mejoran la solución, con respecto a cada una de las técnicas conocidas.

En la sección 2 se indican los operadores básicos y también se presenta la forma paramétrica general de la familia de algoritmos de búsqueda estocástica en paralelo. Para mostrar las bondades de esta nueva técnica, en la sección 3 se presenta un ejemplo de aplicación para bajar el peso de una estructura articulada de una nave industrial sujeta a cargas laterales. Finalmente se pueden encontrar las conclusiones en la sección 4.

2. 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, 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 ordenada de vectores (x_1, \dots, x_N) , $x_i \in \Omega$ sobre los que se aplican continuamente cuatro 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.

2.1. Operadores

2.1.1. 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 de Residuos Estocásticos* 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.

Operador $S_\gamma(X)$	
Calcula	para cada elemento x_k de X , la adaptación relativa f_k y 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 al azar 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	

Debido a su simplicidad y eficiencia, se seleccionó la *Técnica de Residuos Estocásticos*, cuyo objetivo principal es colocar de forma determinística en la nueva población un número 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 de adaptación que mide la bondad f_γ esta parametrizada pr γ , la cual 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:

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

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

Es posible también utilizar la forma exponencial de la forma [4]:

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

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.

2.1.2. Mutación

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

Operador $M_\mu(X)$	
<p style="margin-left: 40px;">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 de Q_i con una probabilidad uniforme</p>	<p style="margin-left: 40px;">Hacer</p>
$M_\mu(X) = (y_1, \dots, y_N)$	
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 adaptación de x . Podemos escribir la probabilidad adaptativa como [14]:

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

donde f_{\max} y \bar{f} son el valor máximo y promedio de la función de adaptación 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.

2.1.3. 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. Con el fin de imponer diversidad se utiliza cada elemento una sola vez en el operador de cruzamiento y se debe garantizar que no se pierda el mejor de todos ellos.

Operador $C_\xi(X)$	
Si bien algunos elementos de X no se han utilizado en cross-over:	
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	

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 adaptación 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.

2.1.4. Aceptación

Finalmente, para realizar la selección entre una población X y un candidato de la población mutada Y se aplica el criterio de aceptación de 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:

2.2. Algoritmo General de Búsqueda Estocástica

2.2.1. Definición

El Algoritmo General de Búsqueda Estocástica (AGBE) se define por el sistema dinámico:

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

Con

$$Y^{(t)} = S_\gamma(X^{(t)}) \quad (5)$$

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. $\exp[-\beta\Delta U]$
x_k con prob.	$(1 - \exp[-\beta\Delta U])$
Hacer	$A_\beta(X, Y) = (u_1, \dots, u_N)$
Algoritmo 4	

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:

1. Para Recocido Simulado (RS) Clásico con N puntos de inicio: $\xi = \gamma = 0$; β incrementándose;
2. Estrategia Evolutiva: $\beta = \xi = 0$;
3. Algoritmo Genético: $\beta = 0$;
4. Recocido Simulado Paralelo Recombinado : $\gamma = 0$; β incrementándose

Se pueden encontrar valores apropiados diferentes de cero para todos los parámetros con el fin de incrementar la eficacia de cada uno de estos algoritmos.

2.2.2. Propiedades de Convergencia

Es fácil ver que si el operador de mutación puede alcanzar algún estado partiendo de cualquier otro, con probabilidad positiva, el algoritmo encontrará un óptimo global con probabilidad uno.

Si este no es el caso, se deben imponer restricciones sobre la velocidad de crecimiento de β hacia el infinito. En el RS clásico, la prueba de convergencia está dividida en dos partes [9]: en la primera se muestra que para una β fija la serie converge a una distribución de Gibbs y en seguida, por medio de las conocidas propiedades de las series de Markov inhomogéneas, se muestra que es posible calcular su límite cuando β tiende a infinito. En nuestro caso, esta estrategia de prueba falla debido a que la distribución de equilibrio de la cadena de Markov que produce el algoritmo general no es Gibbsiana. Sin embargo, puede demostrarse que el algoritmo encontrará el óptimo global (con probabilidad uno) si y solo si el parámetro β crece con respecto al número de iteración n a una velocidad menor que $C \log n$ donde C es una constante. Los detalles de esta demostración y una caracterización asintótica de la distribución de estados de equilibrio para el AGBE se encuentra en [15].

3. Aplicación a la Optimización de una Estructura Plana de Barras Articuladas

Estudiaremos el comportamiento del AGBE para un problema "real": la optimización de estructuras planas articuladas [6] 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 los esfuerzos 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:

$$U(x) = \sum_e \left[\rho_{x(e)} A_{x(e)} L_e + \lambda (\delta \sigma_e) \right] \quad (10)$$

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 esfuerzo que excede el valor máximo permisible y λ es un factor de penalización para el esfuerzo en exceso (en nuestro caso consideramos $\lambda = 1000$).

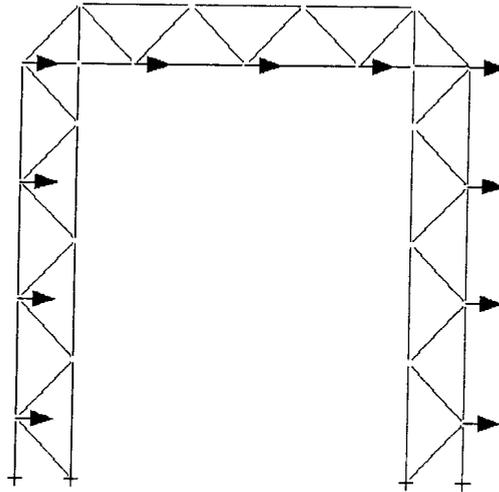
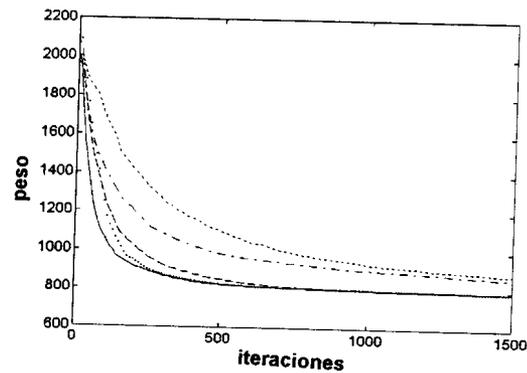
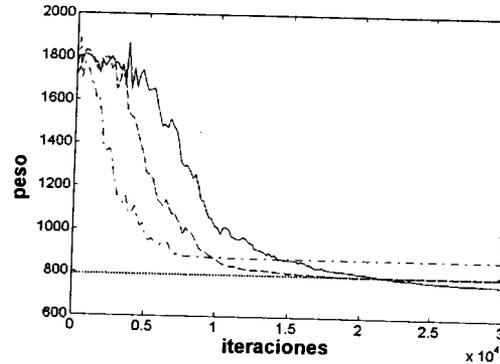


Figura 1. Estructura de barras articuladas de una nave industrial utilizada para encontrar su peso mínimo cumpliendo condiciones de servicio.

En la figura 1 se muestra la estructura articulada estudiada, 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 $2.1 \times 10^6 \text{ Kg/cm}^2$ y el máximo esfuerzo permisible es de $3,500 \text{ Kg/cm}^2$. Las secciones



(a)



(b)

Figura 2. Resultados de la optimización en peso de la estructura de nave industrial de la Figura 1. (a) Comportamiento de la convergencia (sobre 20 corridas de Monte Carlo) 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).

transversales posibles van de 0.1 hasta 50.1 cm^2 con incrementos de 0.5 cm^2 , 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.

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 distintas 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 utilizando RS.

4. Conclusiones

Se presenta una familia de búsqueda estocástica en paralelo de 4-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 análisis de esta técnica indica que la distribución invariante de cadena de Markov asociada es no Gibbsiana y tiene las siguientes propiedades:

1. Si se utiliza la misma secuencia de temperaturas que en el caso de Recocido Simulado, se encuentra el óptimo global de la correspondiente función de costo con probabilidad uno.
2. Asintóticamente, la probabilidad de encontrar el mínimo global en la población es mayor que la correspondiente a la técnica de recocido simulado a la misma temperatura.

Este tipo de algoritmos de familias paramétricas, 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 del espacio de parámetros.

Se realizó un número limitado –pero consideramos, importante – de ejemplos, de los cuales extraemos las siguientes conclusiones:

Los mejores resultados se obtuvieron para técnicas de Recocido Simulado en Paralelo con Interacción, ya sea por medio de 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 selección [15], 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 mejora a medida que el tamaño de la población se incrementa; sin embargo la carga computacional total también se aumenta. El mejor resultado puede ser obtenido utilizando máquinas paralelas donde el número de procesadores sea igual al número de poblaciones, de tal forma que un procesador es asignado a cada individuo. 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 solo procesador.

Finalmente, se puede considerar extender el algoritmo con más operadores y parámetros. Un candidato es incluir la diversidad de forma explícita. Aunque en algunos casos puede ser posible obtener mejores resultados, si exceptuamos el caso en que se desean obtener todos los óptimos [11], 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** and **A. Federgruen**, "Simulated annealing methods with general acceptance probabilities" *J. Applied Prob.* 24 pp 657-667 (1987).
- [3] **T. Boseniuk** and **W. Ebling**, "Boltzmann-, Darwin- and Haeckel-strategies in optimization problems" *Lecture Notes in Computer Science: Parallel Problem Solving from Nature* 496, 430-444 (1991).
- [4] **M. de la Maza** and **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** and **M.J. Walsh** *Artificial Intelligence through Simulated Evolution* New York, Wiley Pub. (1966).
- [6] **M. Galante** "Un algoritmo genético simple para la optimización de estructuras planas articuladas" *Rev. Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* 9 (2) 179-199 (1993).
- [7] **D.E. Goldberg** "A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing" *Complex Systems* 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** and **M.P. Vecchi** "Optimization by simulated annealing" *Science* 220 (4598) 671-680 (1983).
- [10] **F.T. Lin**, **C.Y. Kao** and **C.C. Hsu**, "Incorporating genetic algorithms into simulated annealing" *Proc. of the Fourth Int. Symposium on Artificial Intelligence* 290-297 (1991).
- [11] **S.W. Mahfoud** *Niching methods for Genetic Algorithms* Doctoral dissertation, University of Illinois (1995).
- [12] **S.W. Mahfoud** and **D.E. Goldberg**, "Parallel recombinative simulated annealing: a genetic algorithm" Tech. Report, Depr. of Computer Science, Univ. of Illinois (1994).
- [13] **I. Rechenberg** *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution* FrommanHolzboog, Stuttgart (1973).
- [14] **M. Srinivas** and **L.L. Patnaik**, "Adaptive probabilities of crossover and mutation in genetic algorithms" *IEEE Trans. on Syst., Man and Cyb.* 24 (5) 656-667 (1994).
- [15] **J.L. Marroquin**, **S. Botello** and **J. V. Horebeek** "A family of parallel stochastic search algorithms". Internal technical Report CIMAT (1995).