# Breve Estudio sobre la Aplicación de los Algoritmos Genéticos a la Recuperación de Información

**O. Cordón[1], F. Moya[2], M.C. Zarco[3]**

[1] Dpto. Ciencias de la Computación e I.A. Univ. de Granada. Ocordon@decsai.ugr.es

[2] Dpto. Biblioteconomía y Documentación. Univ. de Granada. felix@goliat.ugr.es

[3] Facultad de Biblioteconomía y Documentación. Univ. de Granada. mczarco@garfio.ugr.es

**Resumen:**

En este trabajo repasaremos las distintas propuestas existentes en la literatura especializada en el marco de la aplicación de los Algoritmos Genéticos a la Recuperación de Información, describiendo varias de las aplicaciones concretas y analizando los resultados obtenidos y la problemática encontrada.

**Palabras clave:** Sistemas de Recuperación de Información, Algoritmos Genéticos.

**Abstract**:

In this work, the different proposals found in the specialised literature for the application of Genetic Algorithms to the field of Information Retrieval will be reviewed. Some of the specific algorithms will be described, and the obtained results will be analysed along with the existing problems.

**Keywords:** Information Retrieval Systems, Genetic Algorithms.

## 1. Introducción

En los últimos años, se ha experimentado un interés creciente en la aplicación de técnicas de Inteligencia Artificial (IA) al campo de la *Recuperación de Información* (RI) [8] con el propósito de subsanar las carencias de los clásicos Sistemas de RI (SRI) booleanos. En concreto, el paradigma del A*prendizaje Automático*, basado en el diseño de sistemas con capacidad para adquirir conocimiento por si mismos, parece interesante para el área de la RI [1,6,7].

Los *Algoritmos Genéticos* (AGs) [6] no son específicamente algoritmos de aprendizaje, pero ofrecen una metodología de búsqueda potente e independiente del dominio que puede ser empleada en muchas tareas de aprendizaje. Debido a ello, la aplicación de los AGs a la RI se ha incrementado en los últimos años. Entre otros, los AGs se han aplicado en la resolución de los siguientes problemas:

- *Indización de documentos* mediante el aprendizaje de los términos relevantes para describirlos [2] y de sus pesos [14].
- *Agrupamiento* (*clustering*) *de documentos* [3,10] *y términos* [11].
- *Mejoras en la definición de consultas*: aprendizaje automático de los términos de la misma [1], de los pesos de los términos proporcionados previamente por el usuario [9,12,15] o de la composición completa de la consulta incluyendo los términos y los operadores booleanos [13], o los dos anteriores y los pesos de los términos [5].

En este trabajo repasamos el uso de los AGs en el área de la RI, describiendo varias de las aplicaciones existentes y analizando los resultados obtenidos y la problemática encontrada. Por motivos de espacio, no describiremos todas las aproximaciones citadas en cada grupo.

## 2. Introducción a los Algoritmos Genéticos

Los AGs [9] trabajan sobre una población de individuos, *cromosomas*, que codifican posibles soluciones al problema y adaptan dicha población en busca de soluciones mejores mediante un proceso evolutivo basado en selección natural (en forma de un proceso de *selección* probabilístico en el que las mejores soluciones se reproducen con mayor probabilidad que las peores) y en la aplicación de operadores genéticos, *cruce* y *mutación*, que modelan los procesos genéticos existentes en la naturaleza. La *función de adaptación* es la encargada de modelar el entorno, definiendo la calidad de las soluciones según como resuelven el problema.

**Procedimiento Algoritmo Genético**

$t \leftarrow 0$. *Iniciar* P(t). *Evaluar* P(t)

*Mientras* no se de la condición de parada *hacer*
t ← t+1
*Seleccionar* P(t) a partir de P(t-1)
*Cruzar* y *mutar* P(t)
*Evaluar* P(t)

Existen distintas formas de llevar a cabo la selección de los individuos. La mas habitual consiste en obtener una distribución de probabilidad asociada a los cromosomas, $P_s^i$ (habitualmente dividiendo la adaptación de cada uno entre la suma de la de toda la población), y en asociar dicha distribución a una ruleta, dando más espacio en la misma a aquellos individuos que presenten mayor probabilidad de selección (es decir, a los mejor adaptados). Esta ruleta se gira tantas veces como cromosomas existan en la población, copiando en la población intermedia el cromosoma escogido en cada caso.

Sobre esta población intermedia se aplican los operadores de cruce y mutación. El primero combina las características de dos cromosomas padre para obtener dos descendientes, mientras que el segundo provoca cambios aleatorios en un único individuo para introducir diversidad en la población. Los dos tienen asociada una probabilidad de aplicación, $P_c$ y $P_m$. Los operadores más clásicos (usualmente empleados con cromosomas binarios) son el cruce simple en un punto (que genera un punto de cruce aleatorio e intercambia los genomas de los dos padres a ambos lados del mismo) y la mutación uniforme (que cambia el valor actual del gen por el valor complementario, 0 por 1 y 1 por 0).

## 3. Aplicación de los Algoritmos Genéticos a la Recuperación de Información

### 3.1. Indización de documentos

Las aplicaciones existentes dentro de este primer grupo están orientadas al aprendizaje mediante adaptación de las descripciones de los documentos existentes en la base documental con objeto de facilitar la recuperación de los mismos ante consultas relevantes.

En [2], *Gordon* defiende que una buena solución para el problema de las diferencias existentes entre consultas de distintos usuarios que buscan los mismos documentos puede ser el asociar mas de una descripción a cada documento y adaptar dichas descripciones a lo largo del tiempo. Para determinar si un documento es o no relevante para una consulta dada, el sistema empareja ésta con todas las descripciones del documento y decide en función del promedio de los emparejamientos parciales.

Así, el autor propone un modelo de AG para llevar a cabo esta tarea. Cada descripción esta compuesta por un vector binario (es decir, sólo se considera la presencia o ausencia de los términos en la descripción del documento actual) de longitud fija. La población genética está formada por diferentes descripciones para el mismo documento (para obtener las descripciones de la colección al completo es necesario ejecutar el AG tantas veces como documentos existan en la misma):

$$C_j = desc_i\text{-}doc_x = (t_{1j}, t_{2j}, \ldots, t_{nj}),\ t_{ij} \in \{0,1\},\ i = 1, \ldots, n$$

Con esta estructura, las distintas descripciones compiten entre sí por representar al documento del mejor modo posible. Para medir la calidad de una descripción, el sistema dispone de dos conjuntos de consultas distintos, para los que el documento es respectivamente relevante e irrelevante. La función de adaptación se basa en calcular la similitud entre la descripción y cada consulta mediante el índice de Jaccard, obteniendo entonces los valores medios de adaptación de la descripción al conjunto de consultas relevantes e irrelevantes. Ambos valores se combinan linealmente, tras invertir el segundo, para obtener una única medida de la bondad de la representación actual del documento.

El AG de Gordon presenta dos características interesantes. Por un lado, no emplea operador de mutación, solamente hace uso de un operador de cruce clásico en un punto para adaptar los individuos. Por otro, dicho operador se aplica sobre toda la población genética (es decir, $P_c$=1).

En lo que respecta al mecanismo de selección, calcula el número de copias de cada cromosoma esperadas en la nueva población dividiendo su valor de adaptación entre la media de la población actual. Automáticamente, cada cromosoma se copia un numero de veces igual a la parte entera de dicha fracción. Los individuos restantes se escogen aleatoriamente de acuerdo a los restos.

Gordon valida el sistema en un entorno real, aunque a pequeña escala, trabajando con una base compuesta por 18 documentos para los cuales diferentes usuarios (estudiantes) proporcionaron una media de 17 descripciones iniciales y consultas relevantes e irrelevantes. Los resultados obtenidos son prometedores. Después de ejecutar el AG durante 40 generaciones, las descripciones de los 18 documentos se asemejaban un 19.09% más a las consultas relevantes y un 24.81% menos a las irrelevantes, es decir, se consigue el objetivo inicial: que las descripciones de los documentos se asemejen en mayor medida a las consultas para las que son relevantes y en menor a las irrelevantes.

En [14], *Vrajitoru* presenta otra aproximación basada en el modelo de espacio vectorial y en cromosomas que codifican la indización de la colección completa de documentos (cada uno de ellos con una única descripción asociada en forma de vector de números reales en [0,1]). La autora propone además un nuevo operador de cruce específico, el cruce desvinculador (*dissociated)*, con el que obtiene buenos resultados sobre dos bases documentales distintas: CACM (3204 documentos y 50 consultas, conociéndose los documentos relevantes) y CISI (1460 documentos y 35 consultas).

### 3.2. Agrupamiento de documentos y términos

*Robertson* y *Willet* [11] proponen un AG para formar grupos (clusters) de palabras de modo que la suma de las frecuencias de aparición de éstas en una colección de documentos sea aproximadamente igual en cada grupo. Este problema tiene varias aplicaciones, como la indización o la generación de signaturas. Para ello, los autores proponen dos esquemas de representación:

1. *Orden con separadores*: Cada individuo es un vector de dimensión N+n-1, donde los números enteros en el conjunto {1, …, N} codifican los términos y n-1 genes con valor –1, los separadores que definen los n clusters. Por ejemplo, con N=8 y n=4, el cromosoma *(1 2 3* ***-1*** *4 5* ***-1*** *6* ***-1*** *7 8)* representa la configuración de clusters: $G_1$ = {1,2,3}, $G_2$ = {4,5}, $G_3$ = {6}, $G_4$ = {7,8}.
2. *División-asignación*: Los cromosomas son vectores de dimensión N. Cada posición está asociada a un término y su valor representa el cluster al que pertenece el termino. Así, el cromosoma *(3 2 1 4 3 3 2 1)* representa la configuración: $G_1$ = {3,8}, $G_2$ = {2,7}, $G_3$ = {1,5,6}, $G_4$ = {4}.

El mecanismo de selección del AG no es el habitual. Cada operador tiene una probabilidad de actuación asociada y, en cada paso, se escoge el operador a aplicar lanzando la ruleta. Una vez que se conoce éste, se escogen el o los padres necesarios para su aplicación del mismo modo. El AG trabaja con dos operadores distintos de mutación (cada uno de los cuales recibe una probabilidad de actuación de 0.25), escogidos de entre tres operadores clásicos para representación de orden [6]. En todos los casos, se escogen primero dos posiciones aleatorias y se efectúa algún cambio entre ellas:

$$C = (1\ 2\ 3 \mid 4\ 5\ 6\ 7 \mid 8)$$

a) *Inversión*: $C' = (1\ 2\ 3 \mid 7\ 6\ 5\ 4 \mid 8)$ (inversión de todos los genes)
b) *Sublista aleatoria*: $C' = (1\ 2\ 3 \mid 6\ 4\ 5\ 7 \mid 8)$ (selección aleatoria de los genes en cuestión)
c) *Posición*: $C' = (1\ 2\ 3 \mid 7\ 5\ 6\ 4 \mid 8)$ (intercambio de los genes de los extremos)

El operador de cruce recibe probabilidad 0.5. Los autores proponen varios, dos de los cuales están también basados en la representación de orden (*cruce ordenado* y *cruce basado en posición*). Para la primera representación, se escoge cualquiera de ellos, mientras que en la representación de división-asignación, los autores consideran además el empleo del cruce simple en un punto y en dos puntos.

La aplicación del c*ruce ordenado* (que será el único descrito por cuestiones de espacio) precisa de la generación aleatoria de un patrón binario con la misma longitud que el cromosoma. Dicho patrón determina si los genes del primer padre se copian al primer hijo (valor 0 en el patrón) o al segundo (1). Los huecos se rellenan con los genes del segundo padre, manteniendo el orden. Por ejemplo:

| | | | | |
|---|---|---|---|---|
| $P = (0\ 1\ 1\ 0\ 1\ 1\ 0\ 0)$ | | *Paso 1:* | | *Paso 2:* |
| $C_1 = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8)$ | ⟶ | $C'_1 = (-\ 2\ 3\ -\ 5\ 6\ -\ -)$ | ⟶ | $C'_1 = (8\ 2\ 3\ 4\ 5\ 6\ 7\ 1)$ |
| $C_2 = (8\ 6\ 4\ 2\ 7\ 5\ 3\ 1)$ | | $C'_2 = (8\ -\ -\ 2\ -\ -\ 3\ 1)$ | | $C'_2 = (8\ 4\ 5\ 2\ 6\ 7\ 3\ 1)$ |

Para la función de adaptación, también se presentan dos propuestas: una *medida de la entropía relativa*, $H_R \in (0,1]$, donde el valor 1 muestra la equifrecuencia total en los clusters, y la *medida de Pratt*, $C \in [0,1]$, donde el valor optimo es el 0. De este modo, el AG afronta un problema de maximización en el primer caso y uno de minimización en el segundo.

Finalmente, en cuanto a los experimentos realizados, los autores trabajan con cinco conjuntos de datos distintos, cuatro de ellos procedentes de distintas colecciones de documentos en ingles y en turco, y el último generado experimentalmente según una distribución de Zipf. Los resultados obtenidos son buenos, aunque similares a un algoritmo existente para el mismo problema que requiere un tiempo de ejecución mucho menor.

Las otras dos aplicaciones incluidas en este grupo son bastante distintas a la anterior. Un problema común cuando se trabaja con procesos de clustering en RI consiste en que el agrupamiento se suele efectuar en función de patrones de similitud en las descripciones de los documentos, lo que da lugar a que documentos que no son relevantes para las mismas consultas pero que presentan descripciones parecidas se agrupen en los mismos clusters, perjudicando la precisión del sistema.

Existen una serie de algoritmos de clustering que solucionan este problema en el campo de la RI, las denominadas *técnicas de agrupamiento orientado al usuario* (*user-based clustering*). En este caso, los documentos se agrupan por su relevancia y no por su descripción, es decir, se busca el formar grupos de documentos relevantes para las mismas consultas. En [3,10] pueden encontrarse dos propuestas basadas en el uso de AGs, realizadas respectivamente por Gordon, y Raghavan y Agarwal.

### 3.3. Definición de consultas

En general, todas las aplicaciones incluidas en este grupo (el más numeroso) tienen en común el empleo de los AGs como técnica de retroalimentación por relevancia (*relevance feedback*) en distintos tipos de SRI. Distinguiremos tres subgrupos dependiendo de los componentes de la consulta adaptados en el proceso genético: los términos, los pesos o la consulta completa (términos, pesos y operadores booleanos).

#### *3.3.1. Aprendizaje de términos*

En [1], *Chen y otros* emplean un AG para aprender los términos de consulta que mejor representan un conjunto de documentos relevantes proporcionados por el usuario (*consulta inductiva a partir de ejemplos* (*inductive query by examples*)). Consideran un SRI basado en el modelo de espacio vectorial en el que las representaciones de los documentos son binarias.

El AG empleado es también un AG binario clásico. Los cromosomas representan posibles conjuntos de términos mediante vectores binarios de tamaño fijo en los que cada posición esta asociada a un termino existente en la colección inicial de documentos relevantes. Los operadores son los habituales en este tipo de AGs (véase la Sección 2).

La función de evaluación tiene que indicar el grado en que el conjunto de términos de consulta actual representa la colección de documentos inicial. Para ello, se considera el coeficiente de Jaccard, de modo se obtiene este valor de similitud entre los términos de consulta actuales y cada uno de los documentos iniciales y se devuelve la media de estos como valor de adaptación del cromosoma:

$$F(C) = \frac{1}{n}\sum_{i=1}^{n} J(C, D_i) \quad ; \quad J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Los autores comparan el comportamiento del proceso genético con otras tres técnicas de retroalimentación por relevancia, una de las cuales es una técnica clásica, expansión de consultas, y las otras dos están también basadas en IA, Simulated Annealing (otro algoritmo probabilístico de búsqueda) e ID3 (un algoritmo para diseñar arboles de clasificación). Hacen uso de una base documental compuesta por 8000 documentos extraídos de la base *COMPEN*, indizando estos mediante las palabras clave que proporciona la propia base, y diseñan dos experimentos distintos: uno simulado con conjuntos iniciales de documentos relevantes de varios tamaños ({1,2,3,4,5,10,20,30,40,50}, 7 conjuntos distintos en cada caso) y uno interactivo con usuarios reales, 21 estudiantes, que proporcionan 36 consultas. En ambos, el AG obtiene los mejores resultados.

#### *3.3.2. Aprendizaje de pesos*

Yen y Korfaghe [15] proponen un AG para retroalimentación por relevancia en SRI basados en el modelo de espacio vectorial mediante el aprendizaje de los pesos asociados a los términos de consulta. Los cromosomas son, directamente, vectores de valores reales en [0,1] y los operadores de cruce y mutación son respectivamente el cruce en dos puntos y la mutación aleatoria.

Por otro lado, el esquema de selección se basa en eliminar todos los cromosomas cuya adaptación sea inferior a la media de la población y en seleccionar la población intermedia aplicando el mecanismo de Baker [6] sobre los supervivientes. La función de adaptación considera las medidas clásicas de precisión y exhaustividad. Para ello, proponen tres funciones distintas, una para el caso en que se conocen todos los documentos relevantes de la base documental para la consulta actual y otras dos para el caso contrario:

$$F_1(C) = R_r - R_n - N_r \qquad F_2(C) = R_r - N_r \qquad F_3(C) = 2 \cdot R_r - N_r$$

donde $R_r$ es el numero de documentos relevantes recuperados, $R_n$ es el numero de documentos relevantes no recuperados y $N_r$ es el numero de documentos irrelevantes recuperados.

Además, las probabilidades de cruce y mutación van cambiando a lo largo de la ejecución del AG con objeto de realizar una mejor búsqueda en el espacio de posibles pesos. En concreto, $P_c$ toma el valor inicial de 0.9 y va descendiendo hasta valer 0.6 al final de la ejecución. En cambio, $P_m$ comienza valiendo 0.001 y aumenta progresivamente hasta situarse en 1 (aunque en las ultimas ejecuciones se reduce bruscamente para no perder las buenas soluciones).

La indización de los documentos en el SRI empleado para los experimentos presenta dos variantes: vectores binarios y vectores de pesos, obtenidos multiplicando la frecuencia del termino en el documento por la frecuencia invertida del primero. La medida de similitud considerada es la distancia y la recuperación se efectúa con umbrales, la raíz cuadrada del numero de términos en la consulta, dividida entre 2 para la indización binaria y entre 1.5 para la indización con pesos.

Para los experimentos realizados consideran la base *CRANFIELD*, sobre la que experimentan el algoritmo con distintos valores de parámetros: 16 tamaños de población distintos, múltiplos de 2 entre 20 y 40, durante 20 generaciones, y con las tres funciones de adaptación presentadas.

Una vez ejecutado el método sobre todas las consultas existentes, miden el nivel de precisión y exhaustividad de cada una, contando el número de ellas existente en cada intervalo de dimensión 0.1, y comparándolo con las consultas iniciales sin pesos. Los resultados son muy satisfactorios, generándose muchas más consultas cercanas al extremo (1,1) tras la adaptación genética. Sin embargo, aunque la precisión media para varios niveles de exhaustividad aumenta claramente, no se observa el mismo resultado en la exhaustividad media para varios niveles de precisión.

Otras aportaciones incluidas en este grupo son la de *Robertson* y *Willet* [12], que proponen un AG con el propósito de determinar un umbral de rendimiento para las técnicas de retroalimentación por relevancia en SRI basados en espacio vectorial, y la de *Sanchez y otros* [9], que proponen un AG para aprender los pesos de los términos en consultas con operadores booleanos en un SRI difuso.

*3.3.3. Aprendizaje de la consulta al completo*

Por último, en este tercer apartado se han propuesto dos técnicas, una para SRI difusos [5] y otra para SRI booleanos [13], ambas basadas en una variante de Algoritmo Evolutivo distinta a los AGs, la Programación Genética (PG) [4]. La principal diferencia entre ambos es el esquema de representación, que en el caso de la PG permite manejar estructuras más complejas tales como árboles de expresiones.

En [5], *Kraft y otros* proponen un algoritmo de PG para aprender la estructura de consultas compuestas por términos ponderados y unidos por medio de operadores booleanos en un SRI difuso. Estas consultas están representadas en forma de árboles y el algoritmo emplea el operador de cruce clásico en PG, basado en seleccionar aleatoriamente una rama del árbol en cada uno de los padres e intercambiar los subárboles que cuelgan de esta [4]. En cambio, para el operador de mutación, consideran tres posibilidades: cambiar un operador booleano por otro, un termino por otro o el peso de un termino, sumándole un valor aleatorio cercano a 0.

El mecanismo de selección considerado no es el generacional habitual, basado en la creación de una población intermedia (véase la Sección 2), sino que está basado en el *esquema estacionario* (*steady-state*). En este caso, en cada generación únicamente se aplica un cruce sobre dos padres y una mutación sobre otro padre distinto, con lo que se obtienen tres descendientes (dos del cruce y uno de la mutación). Estos descendientes pasan a formar parte de la nueva población siempre que estén mas adaptados que los peores individuos de la población actual.

Para la función de evaluación, proponen tres posibilidades, considerar solamente la exhaustividad, la exhaustividad y la precisión y una función para SRI basados en espacio vectorial (Salton [8]):

$$F_1(C) = \frac{\sum_{i=1}^{M} r_i \cdot f_i}{\sum_{i=1}^{M} r_i} \quad ; \quad F_2(C) = \alpha \cdot \frac{\sum_{i=1}^{M} r_i \cdot f_i}{\sum_{i=1}^{M} r_i} + \beta \cdot \frac{\sum_{i=1}^{M} r_i \cdot f_i}{\sum_{i=1}^{M} f_i}$$

$$F_3(C) = \alpha \cdot \sum_{i=1}^{M} \left[ r_i \cdot \sum_{j=1}^{n_{dc}} \frac{sim(i, c_j)}{n_{dc}} \right] - \beta \cdot \sum_{i=1}^{M} \left[ (1 - r_i) \cdot \sum_{j=1}^{n_{dc}} \frac{sim(i, c_j)}{n_{dc}} \right]$$

donde $r_i$ es la relevancia del documento i, $f_i$ es la recuperación del documento i ($r_i$, $f_i \in \{0,1\}$), $n_{dc}$ es el numero de conjunciones que componen la consulta y $sim(i,c_j)$ es el grado de similitud entre la conjunción $c_j$ y el documento i. Hemos de señalar que, para emplear esta ultima función, simplifican el tipo de consulta considerada, pasando a trabajar con vectores de términos unidos por conjunciones. Consideran el coseno como medida de similitud.

En la experimentación hacen uso de una colección con 483 resúmenes de documentos de la base *Comunicaciones de la ACM*, con un total de 4923 términos de indización, sobre la que diseñan dos experimentos. En el primero, un usuario aporta 19 documentos relevantes, mientras que en el segundo se trabaja con 30 documentos, seleccionando uno al azar y añadiendo los 29 mas similares.

Trabajan con las tres funciones de adaptación propuestas y, además, consideran dos posibilidades para generar la población inicial: crearla al completo partir de términos que aparezcan en el conjunto inicial de documentos o generar un 80% de la población con estos términos y el 20% restante con términos que no se encuentren en la misma. Los resultados obtenidos con las dos primeras funciones de evaluación son buenos (18 de los 19 documentos relevantes recuperados en el primer experimento y 27-28 de los 30 en el segundo con $F_1$; y 13-19 (con 2-6 irrelevantes) y 26-27 (con 66-85 irrelevantes) con $F_2$). En cambio, no son capaces de obtener resultados aceptables cuando consideran $F_3$.

Por otro lado, Smith y Smith proponen un algoritmo similar para SRI booleanos [13], en el que las consultas están compuestas por términos unidos por operadores booleanos, sin considerar pesos. Los autores validan el algoritmo sobre la base *CRANFIELD*, buscando la consulta perfecta, y sólo obtienen buenos resultados cuando la colección inicial de documentos relevantes es pequeña.

## 4. Comentarios finales

En este trabajo, hemos repasado las distintas técnicas propuestas en la literatura especializada en el marco de la aplicación de AGs a RI. Tal y como se ha indicado, los AGs se han aplicado fundamentalmente a tres tareas en este campo: *indización de documentos*, *agrupamiento de documentos* y términos y *mejora en la definición de consultas*. En todas ellos, los distintos autores han obtenido resultados adecuados y planteado líneas futuras de trabajo, lo que permite considerar a los AGs como una técnica muy prometedora en SRI.

## Bibliografía


[1] Chen, H. A Machine Learning Approach to Inductive Query by Examples: An Experiment Using Relevance Feedback, ID3, Genetic Algorithms, and Simulated Annealing. *Journal of the American Society for Information Science*, 49(8), 1998, 693-705.

[2] Gordon, M.D. Probabilistic and Genetic Algorithms for Document Retrieval. *Communications of the ACM*, 31(10), 1988, 1208-1218.

[3] Gordon, M.D. User-Based Document Clustering by Redescribing Subject Descriptions with a Genetic Algorithm. *Journal of the American Society for Information Science*, 42(5), 1991, 311-322.

[4] Koza, J.R. *Genetic Programming: On the Programming of Computers by Natural Selection*. MIT Press, 1992.

[5] Kraft, D.H., Petry, F.E., Buckles, B.P., Sadasivan, T. Genetic Algorithms for Query Optimization in Information Retrieval: Relevance Feedback. En: Sanchez, E., Shibata, T., Zadeh, L.A., eds. *Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives*, 1997, 155-173.

[6] Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1996.

[7] Moya, F., Herrero, V., Guerrero, V. La aplicación de las Redes Neuronales Artificiales a la recuperación de la información. *Anuario SOCADI de Documentación e Información* (Barcelona), 2, 1998, 147-164.

[8] Salton, G., McGill, M. *An Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[9] Sanchez, E., Miyano, H., Brachet, J.P. Optimization of Fuzzy Queries with Genetic Algorithms. Application to a Data Base of Patents in Biomedical Engineering. *Actas 6th IFSA World Congress*, Sao Paulo (Brasil), 1995, Vol. II, 293-296.
[10] Raghavan, V.V., Agarwal, B. Optimal Determination of User-oriented Clusters: An Application for the Reproductive Plan. *Actas 2nd Conference on Genetic Algorithms and Their Applications*, Hillsdale, NJ (EEUU), 1987, 241-246.
[11] Robertson, A.M., Willet, P. Generation of Equifrequent Groups of Words Using a Genetic Algorithm. *Journal of Documentation*, 50(3), 1994, 213-232.
[12] Robertson, A.M., Willet, P. An Upperbound to the Performance of Ranked-Output Searching: Optimal Weighting of Query Terms Using a Genetic Algorithm. *Journal of Documentation*, 52(4), 1996, 405-420.
[13] Smith, M.P., Smith, M. The Use of Genetic Programming to Build Boolean Queries for Text Retrieval Through Relevance Feedback. *Journal of Information Science*, 23(6), 1997, 423-431.
[14] Vrajitouru, D. Crossover Improvement for the Genetic Algorithm in Information Retrieval. *Information Processing & Management*, 34(4), 1998, 405-415.
[15] Yen, J.J., Korfhage, R.R. Query Modification Using Genetic Algorithms in Vector Space Models. *International Journal of Expert Systems*, 7(2), 1994, 165-191.