

5. Estructuras del Universo. Editor C. Pajares. Editorial Universidad de Santiago de Compostela (1994)
6. Conversaciones en Compostela. Revista Española de Física 10 (2) 1-3 (1996)
7. C.N. Yang y R. Mills. Phys Rev 95, 631 (1954); 96, 191 (1954)
8. Conferencias del Foro Universitario. P. Ricoer, H. Kung, C. Ayala, C.N. Yang y otros. Ed Univ Santiago de Compostela (1999)
9. C.N. Yang y C.P. Yang. Phys Rev 150, 321 (1966); 150, 327 (1966); 151, 258 (1967)
10. C.N. Yang. Phys Rev Lett 19, 1312 (1967); Phys Rev 168, 1920 (1968)
11. B. Sutherland. Phys Rev Lett 19, 103 (1967)
12. R.J. Baxter. Ann. Phys 70, 193 (1972)
13. C.N. Yang. Phys Rev 85, 808 (1952)
14. C.N. Yang and T.D. Lee. Phys Rev 87, 404 (1952). T.D. Lee and C.N. Yang. Phys Rev 87, 410 (1952)
15. C.N. Yang. Rev of Modern Physics 34, 4, 694 (1962)
16. N. Buyers y C.N. Yang. Phys Rev Lett 7, 46 (1961)
17. Vease la contribución de A. Tonomma en el libro citado en referencia (4)
18. E. Fermi y C.N. Yang. Phys Rev 76, 1739 (1949)
19. T.D. Lee, M. Rosenbluth y C.N. Yang. Phys Rev 75, 905 (1949)
20. T.D. Lee, R. Oehme y C.N. Yang. Phys Rev 106, 340 (1957)
21. T.T. Wu and C.N. Yang. Phys Rev Lett 13, 380 (1964)
22. T.D. Lee and C.N. Yang. Il Nuovo Cimento 10, 749 (1956)
23. T.D. Lee and C.N. Yang. Phys Rev Lett 4, 307 (1960)
24. C.N. Yang. Phys Rev Lett 33, 445 (1974)
25. T.T. Wu y C.N. Yang. Phys Rev D12, 3845 (1975)
26. T.T. Chou and C.N. Yang. Phys Lett B 135, 175 (1984)
- Phys Rev D 32, 1692 (1985). T.T. Chou, C.N. Yang y E. Yen. Phys Rev Lett 54, 510 (1985)
27. C.N. Yang. Phys Rev 77, 242 (1950)

Carlos Pajares

Depto. de Física de Partículas
Universidad de Santiago de Compostela

Optimización Global y Algoritmos Genéticos

1. INTRODUCCIÓN

En buena parte de la Matemática Aplicada es frecuente encontrarse con la situación de tener que optimizar algún tipo de función, para la maximización de un recurso o la minimización de un riesgo. En tales casos, se parte siempre de cierta *función objetivo*

$$f: D \rightarrow \mathbb{R}, \quad (1)$$

planteándose el problema en términos de encontrar algún punto del dominio D , llamémosle x_0 , que cumpla la condición

$$x_0 = \arg \max_{x \in D} f(x)$$

En lo que sigue, y para evitar confusiones, se identificará *optimizar* con *maximizar*. En caso de pretender resolver un problema de minimización, se puede transformar éste en uno de maximización merced a la igualdad

$$\min\{f(x)\} = -\max\{-f(x)\}$$

En muchas situaciones reales la función f es demasiado compleja como para poder ser maximizada con las herramientas del análisis matemático. Algunas complicaciones que se pueden presentar son:

- El dominio D no es un conjunto numérico, sino quizás una familia de grafos representando redes de distribución de transportes.
- La función f puede tener múltiples puntos de discontinuidad.
- La función f no tiene una expresión matemática analítica o algebraica, obteniéndose sus imágenes tras algún proceso de simulación, estocástica o no, que puede dar lugar a la aparición del fenómeno de fractalidad o de no derivabilidad de la función.

Los algoritmos clásicos que se utilizan para maximizar funciones basan su criterio de búsqueda en seguir la dirección de la máxima pendiente (*hillclimbing strategies*). Tienen algunos de ellos la ventaja de no exigir restricciones a la función objetivo, pero su gran desventaja es la de quedarse "atascados" en el primer máximo local que encuentran. Cuando un analista desea maximizar una función, busca el máximo global y difícilmente se conformará con un máximo local o relativo. Así pues, a aquellas situaciones citadas como causantes de aumentar la complejidad del problema planteado, hay que añadir esta otra:

- La presencia de una multiplicidad de máximos locales en los que el algoritmo puede quedar frenado, posiblemente lejos de la mejor solución.

Lo deseable sería disponer de un método que permitiese obtener el máximo global de una función sin exigirle a ésta condiciones difíciles de cumplir. Vaya por delante que tal método no existe, pero la familia de los Algoritmos Genéticos (AG) permite alcanzar soluciones de compromiso, generalmente aceptables, que si bien no aseguran la obtención de los óptimos globales, al menos son capaces de no quedarse frenados en el primer extremo local que encuentran.

2. ALGORITMOS GENÉTICOS

En pocas palabras, los AGs mimetizan la naturaleza cuando ésta actúa de forma que los organismos y especies vivientes se adaptan al medio y evolucionan de manera que su aptitud frente al entorno mejore de una generación a la siguiente como consecuencia de la presión selectiva, provocando la llamada *evolución biológica*. Tanto la acumulación de experiencia por parte del individuo, como su *programa* heredado y almacenado en su código genético, conforman los factores que condicionan su comportamiento frente al entorno. El

proceso de *selección natural* presiona para que los individuos mejor adaptados tengan descendencia y así sus características genéticas se propaguen a las futuras generaciones. Cada individuo guarda en sus *cromosomas*, en forma de *secuencias de genes*, su propio código genético; la reproducción sexual se realiza de forma tal que los cromosomas de los padres se *cruzan* o *recombinan* para generar nuevas secuencias de código. En ocasiones, *mutaciones* debidas a errores en la replicación, a la presencia de radiaciones o de ciertos compuestos químicos pueden alterar algún gen del cromosoma, introduciendo así un factor de diversidad.

La identificación de los conceptos de *adaptación* y *optimización* no es gratuita; si bien algunos autores prefieren separar sus respectivos significados, otros defienden su equivalencia, admitiendo que la única diferencia entre ambos es tener plantados sus orígenes en disciplinas diferentes (Bäck, 1996).

2.1. Codificación

Sea cual sea el dominio D de la función (1), el uso de AGs requiere la codificación de sus elementos en forma de cadenas binarias. Si bien en ocasiones el proceso de codificación es inmediato, en otras exigirá un exhaustivo y detenido estudio previo, basado en el propio contexto en el que aparece el problema de maximización, pues una codificación bien elegida redundará en un procedimiento de optimización más efectivo. Veamos algunos ejemplos:

- En un problema de buscar la conexión entre un conjunto A de ciudades mediante una red de transportes, cada solución posible será un grafo $G \subset A \times A$, de manera que el conjunto de posibles soluciones o dominio sea

$$D = \{G : G \subset A \times A\}$$

Cada grafo $D \in D$ se puede representar por una matriz cua-

drada binaria de orden $\text{Card}(A)$, de forma que el elemento (i, j) -ésimo es igual a 1 si hay línea desde la ciudad i hasta la j , siendo 0 en caso contrario. Con la yuxtaposición de las filas de la matriz se tendrá el cromosoma que codifique al grafo correspondiente.

- Si $D = [a, b] \subset \mathbb{R}$ coincide con un intervalo cerrado de la recta real, dependiendo de la precisión requerida, se puede optar por una representación binaria de L genes, de forma que a se identifique con la secuencia de L ceros y b con la de L unos:

$$a \equiv \underbrace{00\dots0}_{L \text{ ceros}} \quad \text{y} \quad b \equiv \underbrace{11\dots1}_{L \text{ unos}}$$

Si N es el número natural en base decimal correspondiente a $1\dots1_2$ y m el asociado a cualquier otra cadena binaria de longitud L , entonces la solución en $[a, b]$ codificada será

$$x = a + \frac{b-a}{N} \cdot m$$

El número necesario de genes (o longitud del cromosoma, L) para codificar el conjunto de los números reales dentro del intervalo $[a, b]$ depende de la amplitud de éste y de la tolerancia al error o precisión con la que se quiera trabajar; nombrando τ a la tolerancia, el número de genes necesarios viene dado por

$$L = \left\lceil \log_2 \left(1 + \frac{b-a}{\tau} \right) \right\rceil, \quad (2)$$

siendo $\lceil \cdot \rceil$ la función de redondeo.

- Si $D = [a_1, b_1] \times \dots \times [a_d, b_d] \subset \mathbb{R}^d$ es un intervalo cerrado del espacio real euclídeo de dimensión d y la i -ésima componente se codifica con un cromosoma binario de L_i genes, entonces la codificación de cada elemento $x \in D$ requerirá un cromosoma de longitud

$$\sum_{i=1}^d L_i.$$

2.2 Selección

La presión selectiva es la que permite a aquellos cromosomas que se adaptan bien al medio, con un alto valor de la función objetivo, que se reproduzcan para que sus genes o características se propaguen a las siguientes generaciones. Hablando en términos menos biológicos, aquellas soluciones candidatas que mejor se comporten frente a la función objetivo f se combinarán entre sí para generar un nuevo conjunto de soluciones candidatas que heredarán los caracteres de sus antecesores y presumiblemente reforzarán la búsqueda de mejores cromosomas.

Para emular este proceso selectivo, los diseñadores de AGs pueden optar por varias vías:

- Selección de los mejores.** Es la solución más rápida e intuitiva, pero también la más engañosa e ineficiente desde el punto de vista de la optimización global, pues elimina la diversidad y es ciega a la exploración de alternativas.
- Probabilidad proporcional a la adaptación, (roulette wheel).** Si $\{f(c_1), f(c_2), \dots, f(c_n)\}$ son los valores de adaptación asociados a la población de cromosomas $\{c_1, c_2, \dots, c_n\}$, entonces c_i será elegido como progenitor con probabilidad

$$P_i = \frac{f(c_i)}{\sum_{i=1}^n f(c_i)}.$$

Esta opción es válida cuando la función objetivo toma siempre valores positivos, $\text{Im}(f) \subset \mathbb{R}^+$. Éste es el método selectivo que se usó en los primeros AGs (Holland, 1975); es sencillo, pero puede tender a una convergencia prematura bajo la presencia de cromosomas muy buenos frente a otros muy malos. Por otro lado, cuando las diferencias entre las adaptaciones $f(c_i)$ son mínimas, este proceso de selección proporcional a la adaptación casi uniforme, con lo que no existe competencia entre cro-

mosomas y el proceso no es mejor que el de una búsqueda puramente aleatoria.

- c) **Probabilidad proporcional al rango**, (*rank selection*). En este caso se hace una ordenación de los cromosomas según sean sus adaptaciones, de menor a mayor. Una vez dispuestos de esta guisa, la probabilidad de seleccionar el cromosoma que ocupa la posición i -ésima viene dada por

$$P_{(i)} = \frac{2i}{n(n+1)}.$$

Este método evita los inconvenientes citados en el apartado anterior, de forma que las convergencias dejan de ser prematuras o las fluctuaciones interminables (Whitley, 1989).

- d) **Selección por torneo**, (*tournament selection*). Se comienza por elegir al azar dos o más cromosomas, seleccionando el que aporte un mayor valor de la función objetivo. Éste es un método fácil de aplicar en las versiones en paralelo de los AGs (Yao, 1996).

Otros muchos métodos de selección de progenitores se han utilizado y se encuentran dispersos por la literatura; aquí sólo se ha hecho un esbozo de los más usados, que a su vez pueden admitir variantes.

2.3. Recombinación

La recombinación es el operador genético que mezcla los genes de ambos padres para con ellos obtener la dotación genética de los cromosomas descendientes. No siempre dos cromosomas seleccionados como candidatos a padres se recombinan, lo harán con una probabilidad P_c , cantidad que será uno de los parámetros a decidir antes de la ejecución del procedimiento. Pero cuando el cruce se produce, el analista tiene varias opciones:

- a) **Recombinación monopunto**. Es la técnica más sencilla, consistente en seleccionar al azar un punto de la cadena de genes, obtenien-

padre:	1	0	0	1	0	1	1	1	0	1
madre:	1	1	0	0	1	1	1	0	0	1
hijo 1:	1	0	0	0	1	1	1	0	0	1
hijo 2:	1	1	0	1	0	1	1	1	0	1

Figura 1: Recombinación monopunto. Cruce entre el tercer y cuarto gen.

do el cromosoma descendiente "pegando" el primer segmento de uno de los padres con el segundo del otro, tal como se indica en la Figura 1.

- b) **Recombinación multipunto**. Se cruzan dos cromosomas tomando al azar dos o más puntos de ruptura de la cadena, de forma que cada hijo hereda varios segmentos de cada progenitor.
- c) **Recombinación uniforme**. Se ha comprobado que las dos técnicas anteriores pueden producir sesgos indeseables en el proceso de búsqueda de la solución óptima. La recombinación uniforme actúa asignando a cada posición de la cadena descendiente un gen tomado al azar, y con igual probabilidad, del padre o de la madre, de entre los que ocupan la misma posición. La recombinación uniforme no parece tener lugar en la naturaleza (al menos en la única biosfera de la que tenemos noticia), en la cual sí se han podido encontrar casos de hasta ocho puntos de cruce.
- d) **Recombinaciones específicas**. Cuando la codificación de las soluciones de un problema exige imponer restricciones a los cromosomas, el resultado de una recombinación entre dos cromosomas válidos puede dar como resultado un cromosoma codificando una solución no factible. En estos casos el diseño del operador recombinante debe hacerse de forma específica para el problema que se esté abordando.

2.4. Mutación

La mutación consiste en alterar con una probabilidad muy baja cada gen del cromosoma. Esto exige asignar una probabilidad de

mutación P_m , con la cual, en una codificación binaria, un gen con valor 0 pasa a 1 o viceversa. El factor de diversidad introducido por la mutación es tan importante en estos procesos que ciertos algoritmos evolutivos prescinden de la recombinación, que es un operador de búsqueda local, pero no de la mutación, como en aquellos problemas cuya codificación natural lleva a un cromosoma donde cada gen es un número real (Fogel *et al.*, 1966; Rechenberg, 1973). En tales casos cada cromosoma se muta sumándole una variable aleatoria multidimensional de media nula. De todas formas, la importancia y el papel que juegan la mutación y la recombinación en los AGs es un tema de debate dentro del estudio teórico de los AGs.

2.5. Reemplazo generacional

Tras la selección de los padres más aptos y el nacimiento de los nuevos individuos, queda por decidir cómo se configurará la siguiente generación, para de ella seleccionar nuevos padres y repetir el proceso evolutivo. Entre otras, se tienen las siguientes alternativas:

- a) **Reemplazo completo**. Los descendientes sustituyen completamente a la anterior generación. Tiene la desventaja de que al no ser seguro que los hijos mejoren a los padres, se corre el riesgo de perder buenas soluciones por el camino, que pueden tardar en volver a encontrarse.
- b) **Reemplazo elitista**. Se permite que los padres mejor adaptados sobrevivan en la siguiente generación.

Aquí siempre hay que tomar una solución de compromiso; que los mejores individuos sobrevivan de

una generación a la siguiente puede ser bueno para que no se pierdan las buenas soluciones que han ido apareciendo iteración tras iteración. Sin embargo, permitir que incluso los buenos individuos mueran durante el salto generacional puede ser ventajoso para evitar una convergencia prematura y no quedar atrapados en un máximo local.

2.6. Ejemplo de Algoritmo Genético

En esta sección se presenta el procedimiento general de AG y una aplicación sobre un sencillo problema particular. Los parámetros de llamada al procedimiento incluyen el número de individuos que formarán la población durante cada generación, $NumInd$; el número de nacimientos en cada generación, $NumHij$; el total de iteraciones del algoritmo, $TotGener$, que establece el criterio de parada; las probabilidades de cruce cromosómico, P_c , y de mutación de cada gen, P_m . Todavía no existe una teoría general que permita fijar estos parámetros *a priori*; un proceso previo de sintonización (*tuning*), basado en la prueba y error, junto con la intuición del analista, es todo cuanto se puede hacer. Algunos AGs consideran sus propios parámetros como parte del problema, se habla entonces de los Algoritmos Genéticos Adaptativos.

Son variables internas los contadores de bucles, g y h , el vector $Hijo$ que guarda los nuevos individuos, la doble variable $Padres$ que guarda los cromosomas del padre y madre seleccionados para la reproducción, y las variables Pob y Des , que guardan las poblaciones actual de adultos y de descendientes, respectivamente. Por otro lado se usan las rutinas *Iniciar*, *Seleccionar*, *Recombinar*, *Mutar* y *NuevaGeneracion*, cuyos nombres explican por sí mismos sus cometidos. La variable f es global y se refiere a la función que se pretende maximizar.

```

AG = [NumInd, NumHij, TotGener, Pc, Pm]:=
  Pob(0):= Iniciar[NumInd]; (*inicializa primeros individuos*)
  g:= 1; (*inicializa contador de generaciones*)
  while g ≤ TotGener (*comienza bucle generacional*)
    do h:= 1; (*inicializa el contador para nuevos individuos*)
      while h ≤ NumHij (*comienza bucle reproductivo*)
        do Padres:= Seleccionar[Pob(g-1), f];
          Hijo(h):= Mutar[Recombinar[Padres, Pc], Pm];
          h:= h+1; (*fin bucle h*)
        Des(g):= (Hijo(1), ..., Hijo(NumHij));
        Pob(g):= NuevaGeneracion[Pob(g-1), Des(g), NumInd, f];
        g:= g+1; (*fin bucle g*)
      P(TotGener)

```

Figura 2: Seudocódigo del procedimiento AG.

A continuación se estudia la evolución del GA recién descrito para maximizar la función

$$f(x) = 1 + \frac{\cos(x)}{1 + 0.01x^2},$$

cuya gráfica se representa en la Figura 3. Obsérvese la multiplicidad de máximos locales en los que presumiblemente podría quedarse atrapado cualquier algoritmo de búsqueda.

El primer paso a dar se refiere a la codificación de las soluciones candidatas, lo que exige saber el número de genes que habrá de tener cada cromosoma. Para una búsqueda dentro del intervalo $[-40, 40]$ y una precisión $\tau = 0.001$, la expresión (2) nos devuelve una longitud de cromosoma igual a $L = 16$ genes.

Tomando como parámetros de llamada

$$\begin{aligned} NumInd &= 20, NumHij = 20, \\ TotGener &= 50, \\ P_c &= 0.60, P_m = 0.01, \end{aligned}$$

para cada cromosoma hijo sus padres serán elegidos por torneo, el cruce es del tipo monopunto y el reemplazo generacional se hace de forma completa. En el apartado b) de la Figura 3 se observa la evolución de las mejores soluciones de cada generación, obtenidas descodificando los cromosomas que en cada iteración presentaban los máximos valores de la función

objetivo; finalmente, ya en c), se muestra la serie de las medias de valores de la función objetivo correspondientes a los cromosomas de cada generación, las fluctuaciones que se observan son debidas a la presencia de las “malas” soluciones necesarias para mantener la diversidad.

Los parámetros de llamada a la función AG deben ser tales que permitan sobrevivir a los buenos cromosomas, pero al mismo tiempo evitar la convergencia prematura y facilitar la diversidad suficiente como para explorar nuevas regiones del espacio cromosómico.

3. APLICACIONES REALES

Ya se vió en la sección anterior cómo evoluciona un AG en un caso sencillo. Pero el verdadero interés de estos procedimientos consiste en su potencial para la resolución de otras tareas donde no es fácil encontrar una solución por otros medios; por regla general, estas situaciones suelen aparecer en contextos multidimensionales, con una complejidad tal que cualquier intento de resolución por la vía analítica tiene escasas posibilidades de llegar a buen fin. Se citan a continuación, brevemente, algunas de estas situaciones reales:

a) **Problemas combinatorios.** Forman una amplia familia de situaciones en las que el problema viene definido en términos de

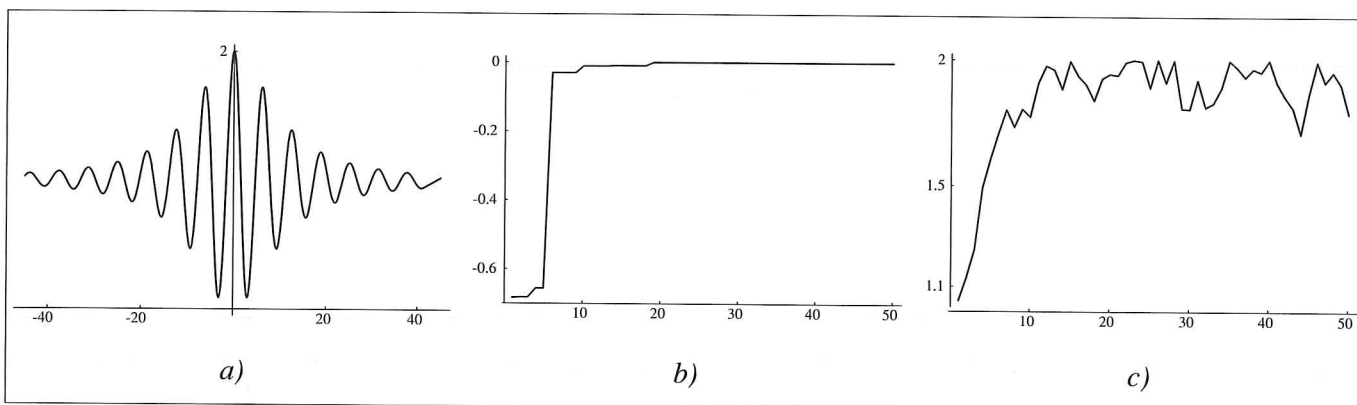


Figura 3: a) gráfico de la función objetivo; b) evolución de las mejores soluciones; c) serie de las medias de las imágenes de la función objetivo.

combinar ciertos entes de forma que haya que buscar aquella situación en la que se optimice el objetivo. Un problema combinatorio clásico es el del *viajante de comercio*, el cual habrá de visitar una serie de ciudades en aquel orden que haga que la distancia total recorrida sea mínima. El problema de *asignación de tareas* se plantea cuando existe una cantidad de trabajos a realizar y una serie de operarios o máquinas que los tienen que llevar a cabo, quizás bajo ciertas restricciones impuestas por convenios laborales o consideraciones de tipo técnico. Un caso particular es la *generación de horarios* en centros de enseñanza; se dispone de una serie de recursos (aulas, gimnasios, laboratorios), de un conjunto de profesores y de otro conjunto de grupos de alumnos; se trata entonces de asignar los recursos, verificando restricciones técnicas y legales.

b) **Diseño de redes neuronales.** Aunque ideadas en los años 40, es desde mediados de los 80 que las *Redes Neuronales Artificiales* se vienen utilizando como paradigmas de cómputo paralelo en el análisis de señales. Clásicamente, la topología que describía la conectividad entre las células se establecía por el analista y los parámetros de la red (los pesos sinápticos) se calculaban por un proceso de entrenamiento. Actualmente, la propia topología de la red neuronal puede obtener-

se vía AGs de forma que optimice la respuesta frente a la entrada de datos.

c) **Estimación estadística.** El ajuste de un modelo paramétrico a un conjunto de datos experimentales debe hacerse, según el criterio que se adopte, maximizando la verosimilitud o minimizando el error que se comete; en todo caso, se trata de un problema de optimización. Algunos modelos pueden ser lo suficientemente complejos como para que los métodos clásicos no alcancen los óptimos buscados, o simplemente no se puedan aplicar a la función objetivo. Los AGs son, una vez más, una alternativa de búsqueda válida.

REFERENCIAS

- Bäck, T. (1996) *Evolutionary Algorithms in Theory and Practice*. Oxford University Press. Nueva York.
- Fogel, L., Owens, A., Walsh, M. (1966) *Artificial Intelligence through Simulated Evolution*. Wiley. Nueva York.
- Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.
- Rechenberg, I. (1973) *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag. Stuttgart.
- Whitley, D. (1989) *The GENITOR algorithm and selective pressure: why rank-based allocation of reproductive trials is best*. Ed. J. Schaffer,

Proc. of the Third International Conference on Genetic Algorithms and Their Applications. 116-121. Morgan Kaufmann. San Mateo, Ca.

Yao, X. (1996) *An overview of Evolutionary Computation*. Chinese Journal of Advanced Software Research. 3, 12-29.

Mario Rodríguez Riotorto*

Doctorando del Depto. de Estadística, Investigación Operativa y Cálculo Numérico
*<http://www.redestb.es/personal/riotorto>

La contaminación de acuíferos por nitratos. Un problema a resolver

Hasta hace relativamente pocos años, la Agricultura ha estado libre de las presiones medioambientales por considerarse actividad no peligrosa desde el punto de vista de la contaminación ambiental. Pero es evidente, que el hombre ha sido capaz de manipular el medio ambiente utilizando diferentes formas de organización y tecnologías, para intentar incrementar la producción de alimentos. Las técnicas intensivas de cultivos, han producido unos innegables beneficios económicos y sociales; sin embargo, de forma paralela, se ha generado una serie de efectos negativos debido tanto a la utilización intensiva de los recursos como a la emisión de contaminantes: la erosión del suelo, la pérdida de nutrientes del mismo o la presencia de sustancias nocivas