

Tema No. 2: Métodos de solución de problemas.

Actividad No. 17

Tipo de clase: Conferencia.

Título: Algoritmos genéticos.

Sumario:



Computación evolutiva

Algoritmos genéticos

Objetivos:



Determinar los elementos fundamentales de la computación evolutiva

Caracterizar los algoritmos genéticos y su aplicación en la inteligencia artificial

Introducción:

Los Algoritmos Genéticos (AG) surgen como herramientas para la solución de complejos problemas de búsqueda y optimización, producto del análisis de los sistemas adaptativos en la naturaleza, y como resultado de abstraer la esencia de su funcionamiento.

Los métodos de búsqueda y optimización han sido objeto de estudio desde los primeros años de la computación extendiéndose desde los métodos basados en el cálculo, pasando por los métodos enumerativos, hasta llegar a los algoritmos de búsqueda aleatoria. Los métodos de búsqueda y optimización tradicionales, los basados en el cálculo, enumerativos y aleatorios puros son analizados y criticados en términos de robustez en [GOL89], ello no significa que no sean útiles; pudiendo servir de complemento a esquemas más robustos para la creación de híbridos.

El término Algoritmo Genético se usa por el hecho de que estos simulan los procesos de la evolución darwiniana a través del uso de operadores genéticos que operan sobre una población de individuos que “evoluciona” de una generación a otra.

El desarrollo de toda la teoría relacionada con el tema no solo ha servido para lograr un método eficiente de búsqueda sino que han permitido abstraer y explicar rigurosamente el proceso adaptativo en los sistemas naturales. Además, esto ha hecho posible diseñar sistemas artificiales que incluyan estos mecanismos naturales.

Algoritmos evolutivos

Los Algoritmos Evolutivos surgieron muy asociados a investigaciones biológicas, especialmente a la simulación del proceso evolutivo.

Los científicos para esto hacían lo siguiente. Definían un arreglo de variables que iba a representar un organismo con sus genes. Según la cantidad de genes que se iban a tener en cuenta, así iba a ser la cantidad de elementos que iba a tener cada arreglo que representa un individuo. Cada uno de los elementos del arreglo toma un valor entre los posibles para ese gen. Por ejemplo, si vamos a representar el ejemplo de los leones, se usaría para representar cada león un arreglo de tres elementos (rapidez, fortaleza, color). Lo otro que tenían que hacer estos científicos era simular la adaptación de los individuos, lo que implicaba crear una función que en función del valor de los genes le diera a cada individuo una probabilidad de subsistir. Para esto pueden crear una función que combine estos valores y pondere los factores, o hacer una simulación del medio natural. Luego

simulando los operadores genéticos y la selección natural van tomando los individuos más aptos y combinándolos, creando nuevas generaciones y ver como evoluciona la especie.

Lo interesante que notaron es que como la función que guía la adaptación la creaban ellos mismos, con esa función gobernaban la evolución. De esta manera, variando la función podrían variar los resultados de la evolución simulada, notando que la evolución se guiaba hacia la mejor combinación de valores de genes, según la función que daba la adaptación.

Por otra parte, en el campo de la Investigación de Operaciones, los científicos estaban buscando maneras de lograr optimizar cualquier tipo de funciones. Estos científicos habían descubierto el método Simplex que permitía optimizar funciones lineales, por ejemplo, buscar el máximo de la función:

$$F(X_1, X_2, X_3) = 3 \cdot X_1 + 18 \cdot X_2 - 3 \cdot X_3$$

Sin embargo, si la función que se quiere optimizar no es lineal (una combinación de coeficientes con variables sin exponentes), entonces el método Simplex no puede usarse.

Por ejemplo, si la función que se quiere maximizar es la siguiente:

$$F(X_1, X_2, X_3) = 3 \cdot \sin(X_1) + 18 \cdot (X_2)^4 + 3 \cdot X_3 \cdot X_1$$

De esta manera, había por una parte un mecanismo que simulaba la evolución que permitía lograr buenas combinaciones de valores para tratar de maximizar cualquier tipo de funciones, y por otra parte había una necesidad de un método de optimización que no tuviera restricciones respecto a las características de las funciones a optimizar.

De esta combinación de necesidades y oportunidades, es que surgen los Algoritmos Evolutivos, y dentro de ellos los Algoritmos Genéticos que son los más populares.

Métodos de búsqueda y optimización

Según David E. Goldberg en su libro "Genetic Algorithm in Search, Optimization and Machine Learning" los AG son: *"algoritmos de búsqueda basados en los mecanismos de la selección natural y la genética"*.

Por tanto lo primero que debe quedar claro es que no están destinados solo a problemas de aprendizaje, aun cuando se reconoce que en este campo han alcanzado cierto desarrollo en los últimos tiempos. Es decir, que aunque se han usado en problemas de aprendizaje, este es sólo un tipo de problemas en los que puede usarse.

El desarrollo de toda la teoría relacionada con el tema **no sólo ha servido para lograr un método eficiente de búsqueda sino que ha permitido abstraer y explicar rigurosamente el proceso adaptativo en los sistemas naturales**. Además, esto ha hecho posible diseñar sistemas artificiales que incluyan estos importantes mecanismos naturales.

Es importante notar que a diferencia de los otros métodos de búsqueda conocidos, como el BestFirst o A* que tienen como propósito encontrar un camino que una un origen con una meta (como es el caso del problema de los caníbales y misioneros, o el de las jarras de agua), en el caso de la búsqueda desde el punto de vista de la optimización, el objetivo es encontrar un punto que no se conoce al inicio, pero que es el que maximiza una función objetivo. En este sentido, su planteamiento es similar a los que se hacen en Investigación de Operaciones, por ejemplo para los problemas de Programación Lineal, donde hay una función a optimizar (maximizar o minimizar) sujeta a determinadas restricciones.

Comparación con otros métodos

La idea de los AG está ligada a la investigación de métodos de búsqueda que superen a los métodos tradicionales, o sea, encontrar un método que sea lo suficientemente robusto como para lograr la eficiencia necesaria para tener un comportamiento bueno en cualquier problema.

A partir de ahora se tratará la búsqueda reduciéndola a la búsqueda del máximo. De forma general todas las búsquedas pueden reducirse a esta variando la función a maximizar.

Los métodos de búsqueda para optimización se pueden clasificar en:

- Métodos basados en cálculos
- Métodos enumerativos
- Métodos aleatorios

Los basados en cálculos pueden ser:

- Métodos Indirectos: Ej. resolver un sistema de ecuaciones y hacer una búsqueda entre los puntos de pendiente igual 0. Se le llama indirecto porque no se trabaja directamente con la función que se desea maximizar sino con una transformación de esta, que en este caso es la derivada.
- Métodos Directos: Ej. moverse de un punto a otro a través del cálculo del gradiente y seleccionando la dirección que tienda al incremento de la función (esto puede verse como una variante del conocido método Hill Climbing o Escalador de Colinas).

Estos métodos pueden necesitar de la existencia de derivadas, por lo tanto limitan un poco el problema a tratar. Además de lo anterior pueden equivocarse y hallar un máximo falso. En el caso del Hill Climbing esto puede verse de la forma siguiente, en que nada hará que el algoritmo salga de este máximo local ya que las pendientes a ambos lados le indica hacia este punto como el más alto. En este caso, mostrado en la figura, después que el escalador siguió el sentido de la flecha y escaló la colina, escogiendo siempre entre los dos posibles movimientos el que lo llevaba al punto de mayor valor, llega al máximo local marcado en el punto rojo. Una vez en esta posición, los puntos a los que puede ir tienen valores menores que la posición actual, por tanto no saldrá del máximo local, y nunca llegará el máximo global.



Los métodos enumerativos se basan en la exploración exhaustiva del espacio. Esto permite que siempre encuentren el máximo pero a costa de mucha demora en la búsqueda. Un ejemplo de esto puede ser la forma tradicional que se usa para buscar el máximo de una lista en Prolog:

$\text{max}([X], X).$

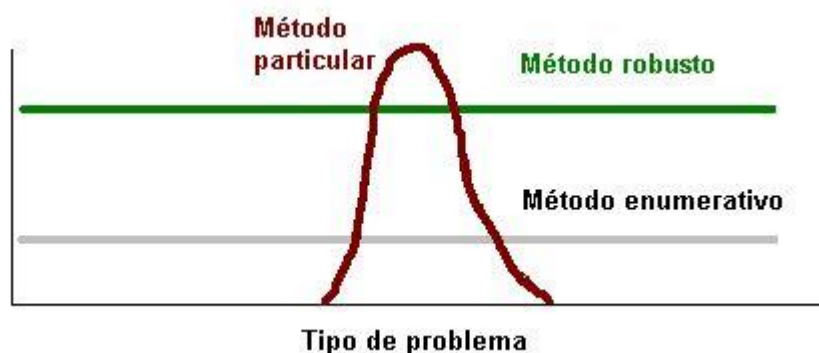
$\max([X|L], Z) :- \max(L, Y),$
 $\text{mayor}(X, Y, Z).$

$\text{mayor}(X, Y, X) :- X \geq Y.$
 $\text{mayor}(X, Y, Y) :- X < Y.$

En estos métodos enumerativos, se emplea la fuerza bruta en busca de la mejor solución, pero cuando las dimensiones de los problemas son demasiado grandes, estos métodos no pueden emplearse.

Por su parte, los métodos aleatorios son los que realizan una búsqueda aleatoria del espacio, sin esto decir que realicen una búsqueda adireccional o sin criterio ni dirección. Estos métodos están ganando popularidad y un buen ejemplo de ellos son los algoritmos genéticos. En esta categoría también se incluyen otros métodos que pertenecen a los algoritmos llamados evolutivos como son las Estrategias de Evolución, la Programación Genética, etc., así como otros métodos también muy populares y que no pertenecen a la familia de los algoritmos evolutivos como son el Recocido Simulado, la Búsqueda Tabú o las Redes de Hopfield.

Un resumen comparativo entre los diferentes métodos se puede ver en el siguiente gráfico. Se puede ver que los métodos enumerativos tienen un comportamiento malo en todos los problemas, sin embargo se puede aplicar. Por otra parte, los métodos particulares (como son los basados en cálculos) tienen un comportamiento muy bueno en algunos problemas, sin embargo, no es posible aplicarlos en otros. Por su parte los métodos robustos tienen un buen comportamiento en todos los problemas, sin llegar a la eficiencia de los métodos particulares en los casos en que estos pueden usarse.



Los AG intentan acercarse al esquema robusto, al igual que los otros métodos aleatorios comentados antes. Las características peculiares de los AG son:

- Trabajan con una codificación de los parámetros y no con los parámetros mismos.
- Buscan a partir de una población de puntos y no de un punto simple.
- Usan directamente la función objetivo y no la derivada u otro conocimiento auxiliar.
- Usan reglas de transición probabilísticas y no determinísticas.

La esencia de su funcionamiento está en el uso de la función objetivo, una codificación de los parámetros, las operaciones con cadenas y el uso de números aleatorios. Se consideran simples y potentes.

Algo de historia

Desde finales de los años 50 y principios de los 60 se desarrollaron trabajos por algunos biólogos para simular sistemas genéticos en una computadora, dentro de estos trabajos es destacable el de

Fraser, entre los años 60 y 62, donde se llegó a algo bastante parecido a los AG, trabajando con cadenas y fenotipos.

A pesar de estos trabajos, se reconoce al profesor John Holland de la Universidad de Michigan como el creador de los AG con su trabajo sobre teoría de los sistemas adaptativos, en el año 62. Es Holland el primero que adaptó la idea de la genética a sistemas artificiales.

El nombre de AG lo usó por primera vez Bagley en el año 67, y el mismo es el autor de un trabajo importante sobre la utilización de AG en juegos.

Holland siguió investigando en este campo y en el año 75 introdujo mejoras importantes en los AG como es el escalado y publicó un libro sobre el tema que es material de referencia clásico [HOL75].

En la actualidad, existen una gran cantidad de problemas en los que se usan los AG y han llegado a consolidarse como métodos de búsqueda y de optimización.

Fundamentos de los AG.

La base fundamental que sustenta la idea de los AG está en la necesidad de combinar ideas para obtener conocimiento nuevo, original y mejor. Esto siempre parte de la base de que tendrán más posibilidad de ser oídas, y por lo tanto combinadas, aquellas ideas que han demostrado ser las más correctas o útiles.

Esta necesidad se sustenta en que si dos ideas han mostrado un comportamiento bueno, entonces la idea que surja de la combinación de ellas debe utilizar los aspectos mejores de ambas de forma conveniente.

Desde el punto de vista biológico el problema se centra en una imitación del mecanismo evolutivo de los seres vivos. De una población siempre tienen más posibilidad de sobrevivir aquellos organismos que mejor adaptados están al medio, como se comentó al principio.

De combinar dos organismos que tengan características deseables para aspectos distintos pueden surgir nuevos organismos que hereden ambas características.

Este proceso es la base de la evolución de los seres vivos y su estudio lo lleva a cabo la Genética.

Terminología usada.

En el trabajo con AG se maneja una serie de términos “importados” de la genética natural. No siempre es adecuada la analogía, pero estos son comúnmente aceptados:

Población	Conjunto de individuos o cromosomas. Equivale a una muestra aleatoria del espacio de solución o un conjunto de soluciones alternativas.
Cromosoma	Un cromosoma es un portador de la información genética que transmite cada uno de sus genes. Una posible solución.
Gen	Cada uno de los rasgos o características que conforman el cromosoma. También se les llama parámetros o aspectos. Cada gen equivale a una variable del problema.
Genotipo	En biología se le llama al “paquete” genético total en su forma interna. En la terminología de AG será la información genética de todo el cromosoma en forma codificada.
Fenotipo	Se le llama en genética al paquete genético tal y como interactúa con el medio exterior. En los AG artificiales serían los aspectos del cromosoma

	decodificados.
Locus	Es la posición de un gen el cromosoma
Alelo	Es el valor asociado a un gen

Formas de trabajo de los AG.

Los algoritmos genéticos se pueden considerar métodos de búsqueda aleatoriamente guiados. Un esquema general de un AG puede ser visto en la **Fig 1**. Se parte de una población inicial que es sometida a un proceso de **selección**, como resultado algunos organismos serán “escogidos” para continuar el proceso evolutivo, consistente en el **cruzamiento** entre ellos y luego la **mutación** individual de cada uno de los organismos resultantes de la etapa anterior. Luego mediante una estrategia de reemplazo estos organismos son incorporados a la población inicial.



Fig 1. Ciclo de evolución

Cada una de estas 3 fases del ciclo puede ser implementada de muchas maneras pero siempre dependiente de la clase de problema que se esté resolviendo.

¿Cómo resolver un problema usando Algoritmos Genéticos? A continuación se enumeran algunos pasos que pueden ayudar a responder esta interrogante.

1. Diseñar una representación
2. Decidir cómo inicializar una población
3. Diseñar una forma de evaluar un individuo
4. Decidir cómo seleccionar los individuos para ser padres
5. Diseñar un operador de cruce adecuado
6. Diseñar un operador de mutación adecuado
7. Decidir cómo reemplazar a los individuos
8. Decidir la condición de parada

Como puede notarse los pasos del 1 al 3 son dependientes del problema, por lo que es muy importante la interpretación del mismo y el estudio de problemas similares (a qué clase de problemas pertenece, e.g. Ordenamiento, Planificación, Tipo mochila, etc).

Los componentes del 4 al 7 son propiamente algorítmicos pero que pueden diseñarse apoyándose en el tipo de problema que se está resolviendo.

El paso 8 estará determinado por el propósito y/o objetivo perseguido con la solución y condicionado por los parámetros de eficacia y eficiencia que se deseen obtener en la solución. Por ejemplo: detener el ciclo solo cuando se encuentre la solución óptima, al cabo de 5 minutos de cálculo, luego de N iteraciones.

Modelos de algoritmos genéticos.

Existen dos modelos fundamentales de AGs: **Modelo Generacional** y **Modelo Estacionario**. En el **Modelo Generacional** durante cada iteración se crea una población completa con nuevos individuos, la nueva población reemplaza directamente a la antigua. Ver **Fig 2**.

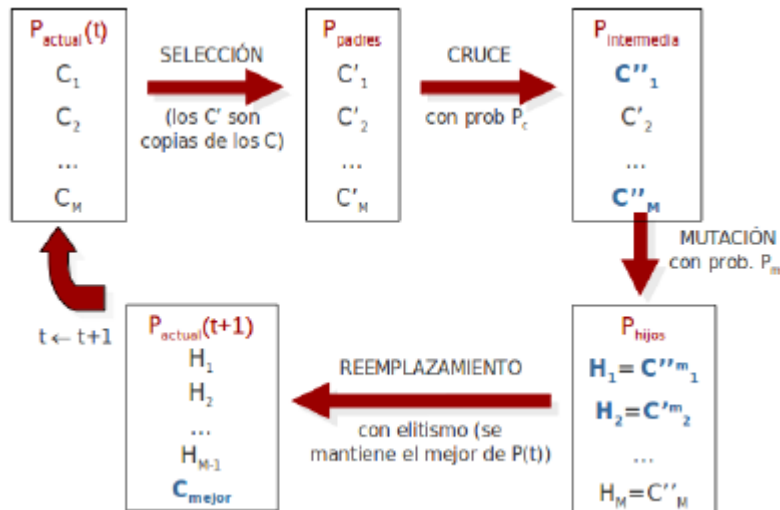
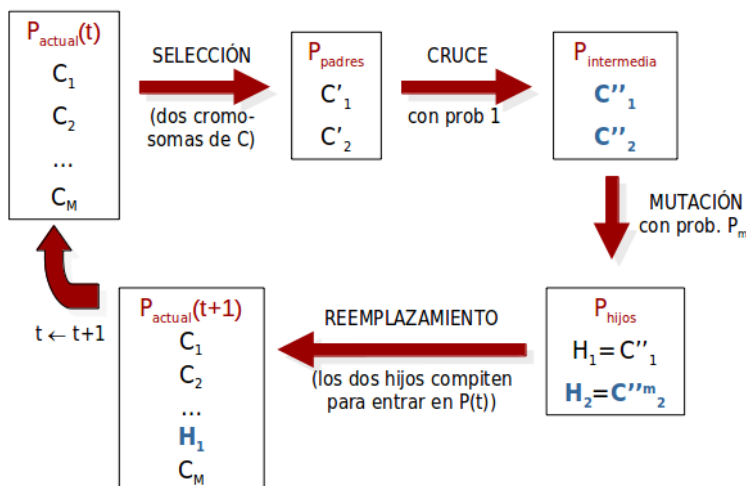


Fig.2 Modelo generacional.

En el **Modelo Estacionario** durante cada iteración se escogen dos padres de la población (diferentes mecanismos de muestreo) y se les aplican los operadores genéticos. El/los descendiente/s reemplaza/n a uno/dos cromosoma/s de la población inicial. Ver **Fig 3**. El modelo estacionario es elitista. Además, produce una presión selectiva alta (convergencia rápida) cuando se reemplazan los peores cromosomas de la población



Estrategias de reemplazo.

Veamos qué estrategias de **reemplazo** pueden encontrarse en la literatura especializada. La presión selectiva se ve también afectada por la forma en que los cromosomas de la población son reemplazados por los nuevos descendientes. Podemos utilizar métodos de reemplazamiento

aleatorios, o determinísticos. Podemos decidir no reemplazar al mejor cromosoma de la población: Elitismo (el uso del Elitismo es aconsejado en los modelos generacionales para no perder la mejor solución encontrada). Un modelo con alto grado de elitismo consiste en utilizar una población intermedia con todos los padres (N) y todos los descendientes y seleccionar los N mejores. Esto se combina con otras componentes con alto grado de diversidad.

Cuando se considera un modelo estacionario (en el que se reemplazan solo uno o dos padres, frente al modelo generacional en el que se reemplaza la población completa), nos encontramos con diferentes propuestas.

A continuación presentamos algunas posibilidades:

1. **Reemplazar al peor de la población (RW)**. Genera alta presión selectiva.
2. **Torneo Restringido (RTS)**: Se reemplaza al más parecido de entre w ($w=3, \dots$). Mantiene una cierta diversidad.
3. **Peor entre semejantes (WAMS)**: Se reemplaza el peor cromosoma del conjunto de los w ($w=3, \dots$) padres más parecidos al descendiente generado (seleccionados de toda la población). Busca equilibrio entre diversidad y presión selectiva.
4. **Algoritmo de Crowding Determinístico (DC)**: El hijo reemplaza a su padre más parecido. Mantiene diversidad.

Cuando se resuelve un problema a partir de AG ...

Veamos a continuación algunos aspectos importantes a tener en cuenta cuando se emplea AG para resolver un determinado problema.

Nunca sacar conclusiones de una única ejecución, esto es utilizar medidas estadísticas (medias, medianas,...) con un número suficiente de ejecuciones independientes. "Se puede obtener lo que se desea en una experimentación de acuerdo a la dificultad de los casos utilizados", no se debe ajustar/chequear la actuación de un algoritmo sobre ejemplos simples si se desea trabajar con casos reales.

Desde el punto de vista de las aplicaciones pueden tomarse dos caminos (doble enfoque y diferente diseño): encontrar una solución muy buena al menos una vez o encontrar al menos una solución muy buena en cada ejecución. Estos criterios evidentemente dependen de las condiciones con que se cuente para la ejecución del AG.

Otros aspectos importantes en los algoritmos genéticos es referido a la **diversidad**, la **exploración** y la **explotación**. El primero de ellos se refiere a la diferencia entre los cromosomas de la población; la **explotación** está referida a cuán bueno es el algoritmo explorando todo el espacio de búsqueda, o sea que sea capaz de no caer en mínimos locales; mientras que la **exploración** se refiere a la capacidad del algoritmo para mejorar las soluciones más prometedoras encontradas durante su ejecución.

En los AG un elemento importante para conseguir una profunda **exploración** del espacio de búsqueda, es el uso de mecanismos que mantengan una determinada separabilidad espacial entre las soluciones de una población tanto para problemas discretos como continuos. Este elemento se conoce como **diversidad** de una población.

La falta de **diversidad** genética es poco deseable pues implica una rápida convergencia al vecino más cercano (alta presión selectiva). En la práctica este parámetro puede controlarse mediante:

1. Inclusión de mecanismos de diversidad en la evolución,
 - Mantener en paralelo varias sub-poblaciones procesadas cada una por un algoritmo genético (Muhlenbein et al., 1991).
 - Evolución con un método de clustering que puede reducir el costo total de la Búsqueda Local evitando redescubrir varias veces los mismos óptimos locales (Seront and Bersini, 2000).

- Mecanismo de Prevención de Incesto, donde se cruzan solo aquellos individuos que presenten una determinada diversidad (Eshelman, 1991).
2. Reinicialización cuando se produce convergencia prematura.

Una excesiva **exploración** conlleva a una búsqueda aleatoria y en consecuencia a la no convergencia, mientras que una excesiva **explotación** se centra en una búsqueda local y por consecuencia convergencia a un óptimo local.

Importancia en la exploración gruesa de grandes espacios

El hecho de explorar varias posibilidades en una sola corrida le permite explorar rápidamente espacios grandes llegando a conclusiones generales sobre las mejores zonas lo cual hace más rápida la exploración de espacios grandes. Esta es una de las principales ventajas sobre otros métodos más exactos pero más lentos.

Uso en problemas donde no son aplicables otros métodos

De lo visto antes puede deducirse que los AG son muy útiles en los problemas donde no existe un algoritmo conocido para solucionarlo.

El AG da una respuesta razonablemente buena donde otros métodos no pueden operar. Esta característica del AG, la comparte con los otros métodos aleatorios y robustos.

Por otra parte no es recomendable su uso en problemas donde ya existe un algoritmo conocido y eficiente que lo resuelve, es decir donde existe un método particular como se vio en el esquema que se mostró en la clases. Por ejemplo, en un problema de Programación Lineal, donde es aplicable el método Simplex, o el caso en que se puedan calcular derivadas no tiene sentido usar un AG.

Posibilidad de combinación con otros métodos para explotar las conveniencias de cada uno (Hill Climbing, Redes Neuronales)

Como se ha visto los AG no caen en mínimos locales, pero sin embargo no garantizan la convergencia al óptimo. Este contraste ha conducido a la combinación con otros algoritmos que como el Hill Climbing o Escalador de Colinas (EC) se pierden en un máximo local pero que cuando están en la vecindad del máximo global entonces sí convergen a la solución.

Lo que se ha hecho es responsabilizar al AG con hallar la zona o colina más alta (vecindad del óptimo) y luego que el Hill Climbing o Escalador de Colinas (EC) sea encargado de escalar hasta la cima. En este caso se ha hablado del Hill Climbing pero pudiera ser cualquier algoritmo basado en gradientes como es el algoritmo de aprendizaje de las Redes Neuronales de Retropropagación.




Esto se muestra en la siguiente figura.



Aplicaciones de los algoritmos genéticos.

1. Optimización combinatoria y en dominios reales
2. Modelado e identificación de sistemas
3. Planificación y control
4. Ingeniería
5. Vida artificial
6. Aprendizaje y minería de datos
7. Visión por Computador e Informática Gráfica
8. Internet y Sistemas de Recuperación de Información

Conclusiones:

-  Los algoritmos genéticos se han venido desarrollando desde hace varios años, como una herramienta para los sistemas inteligentes.
-  Los algoritmos genéticos emplean como principio de funcionamiento las leyes de la selección natural.
-  Los algoritmos presentan diferencias con los métodos tradicionales de solución de problemas de la I.A.

Bibliografía:

Goldberg, David E.: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.