

31

UN ENFOQUE DIDÁCTICO PARA EL ESTUDIO DEL ALGORITMO GENÉTICO EN LA ESTIMACIÓN DE PARÁMETROS DE MODELOS MATEMÁTICOS.

A DIDACTIC APPROACH TO THE STUDY OF GENETIC ALGORITHM IN THE ESTIMATION OF PARAMETERS OF MATHEMATICAL MODELS.

María Alejandra Mármol Martínez¹
Eduardo Ibarguen Mondragón²
Mawency Vergel-Ortega³

Colombia

¹ Departamento de Matemáticas y Estadística, Universidad de Nariño, Pasto, Colombia. Correo: mariamarmol@udenar.edu.co Orcid: <https://orcid.org/0000-0003-4790-0674>

² Departamento de Matemáticas y Estadística, Universidad de Nariño, Pasto, Colombia. Correo: edbargun@udenar.edu.co Orcid: <https://orcid.org/0000-0001-6308-1344>

³ Departamento de Matemáticas y Estadística, Universidad Francisco de Paula Santander, Norte de Santander, Cúcuta, Colombia, correo: mawencyvergel@ufps.edu.co Orcid: <https://orcid.org/0000-0001-8285-2968>

RESUMEN

El presente artículo aborda de manera teórica y global los conceptos básicos alrededor de los algoritmos genéticos, el vocabulario propio de los mismos y un breve análisis del tipo de problemas que se pueden abordar empleando este método de búsqueda. Posteriormente presenta de manera muy general los pasos para realizar la implementación de un algoritmo genético en un software, acompañado de un ejemplo práctico donde se explica detalla y didácticamente línea por línea el código empleado en la programación, para resolver un problema puntual. Un aspecto en el que se enfatizó fue la búsqueda de diferentes aplicaciones de este algoritmo ya que esto permite al lector evidenciar el espectro de problemas que se pueden resolver mediante este método de optimización. Además, de obtener ideas en cuanto a la formulación de la función objetivo, la programación del código, la selección del software, la acotación de parámetros entre otros. En conjunto la información contenida en este artículo busca dar al lector una base de partida para la comprensión, entendimiento, implementación y aplicación de los algoritmos genéticos.

PALABRAS CLAVE: Optimización, fenómeno biológico, operadores, algoritmo genético, método de búsqueda, estimación de parámetros

ABSTRACT

This article addresses the basic concepts around genetic algorithms, their own vocabulary and a brief analysis of the type of problems that can be faced using this search method. Subsequently, we present in a very general way the steps to carry out the implementation of a genetic algorithm through a practical example in which the code used in programming is explained in detail and didactically line by line, to solve a specific problem. One aspect in which we emphasized was the of problems that can be solved by this optimization method. In addition, to obtain ideas regarding the formulation of the objective function, programming of the code, selection of the software, dimensioning of parameters, among others. It is established that the information contained in this article seeks to give the reader a starting point for the compression, understanding, implementation and application of genetic algorithms.

KEYWORDS: Optimization, biological phenomenon, operators, genetic algorithm, search method, parameter estimation

INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS

¿Qué son los algoritmos genéticos?

Los algoritmos genéticos son algoritmos de búsqueda basados en los mecanismos de selección natural y evolución genética. Combinan un fenómeno biológico como es la supervivencia del más apto con un intercambio de información estructurado, que se asemeja a los cambios de información genética presentes en las especies, dicho cambio es aleatorio para formar un algoritmo de búsqueda con algo del estilo innovador de la búsqueda humana. Aunque aleatorios, los algoritmos genéticos no son una simple búsqueda arbitraria, se debe conocer conscientemente la información histórica para inferir sobre nuevos puntos de búsqueda con las características esperadas (Goldberg, 1989).

Ejemplo 1:

El principio fundamental de los algoritmos genéticos es recrear los fenómenos presentes en

la naturaleza. Tomemos el siguiente ejemplo de (Michalewicz, 1999) una población específica de conejos. En un momento dado existe una población de conejos. Algunos son más rápidos e inteligentes que otros conejos, a estos conejos más inteligentes es menos probable que sean cazados por los zorros y, por lo tanto, más de ellos sobreviven. Por supuesto, algunos de los conejos menos rápidos y menos inteligentes pueden sobrevivir solo porque tienen suerte, es decir, un hecho aleatorio. Esta población superviviente de conejos comienza a reproducirse. Los descendientes presentan una buena selección del material genético (aptitudes para sobrevivir a los zorros) de sus progenitores. También ocurre que algunos conejos menos rápidos se reproducen con conejos rápidos, algunos rápidos con rápidos, otros conejos inteligentes con conejos menos inteligentes, y así sucesivamente y puede suceder que en la naturaleza aparezca una liebre salvaje con un gen compatible con los conejos, que muta algunas de las características genéticas del material del conejo. Los conejos bebés resultantes serán (en promedio) más rápidos e inteligentes que estos en la población original porque estadísticamente más padres inteligentes sobrevivieron a los zorros. (Simultáneamente los zorros están pasando por un proceso similar de lo contrario, los conejos podrían volverse demasiado rápidos e inteligentes para que los atrapen cualquiera de ellos).

En el **ejemplo 1**, la población inicial está formada por conejos más y menos rápidos, más y menos inteligentes en diferentes proporciones que por el momento no son trascendentes para el análisis. Por el proceso de selección natural los conejos menos inteligentes y menos rápidos, tienen mayor probabilidad de ser atrapados por un zorro, sin embargo, bajo ciertas circunstancias atribuidas al azar los conejos menos rápidos y menos inteligentes podrían sobrevivir. Lo anterior implicaría que los conejos más inteligentes y más rápidos son los más aptos para sobrevivir generando una nueva población de conejos donde la mayor parte de individuos son más rápidos y más inteligentes. Se esperaría que en la próxima generación de conejos la subpoblación de conejos más rápidos y más inteligentes tenga una mayor proporción con respecto a la primera población.

Ventajas y desventajas de la aplicación de los algoritmos genéticos

Para (Jurado, 2008) se entiende por robustez una medida de la capacidad que tiene el método de permanecer inalterado por pequeñas variaciones en el procedimiento del mismo. Esto proporciona, una indicación de su confiabilidad.

En cuanto a la robustez de la optimización tradicional y métodos de búsqueda se pueden identificar tres tipos de métodos de búsqueda: basados en cálculo, enumerativos o aleatorios. Los algoritmos genéticos están clasificados como un método de búsqueda aleatorio.

Métodos basados en el cálculo: Estos se subdividen en dos clases principales: indirecto y directo. Los métodos indirectos buscan extremos locales resolviendo el conjunto de ecuaciones generalmente no lineal que resulta de establecer el gradiente de la función objetivo igual a cero. Esta es la generalización multidimensional de la noción de cálculo elemental de puntos extremos, como se ilustra en la figura 1. Dado una función suave y sin restricciones que encuentra un posible pico comienza restringiendo la búsqueda de aquellos puntos con pendientes de cero en todas las direcciones. Por otra parte, los métodos directos (de búsqueda) buscan óptimos locales saltando sobre la función y moviéndose en una dirección relacionada con el máximo local (Macías, 1973). Esta es simplemente la noción de escalar una colina para encontrar la mejor zona, subir la función en la dirección más empinada posible. Si bien ambos métodos basados en cálculo se han mejorado, extendido, y repetido algunos razonamientos simples muestran su falta de robustez.

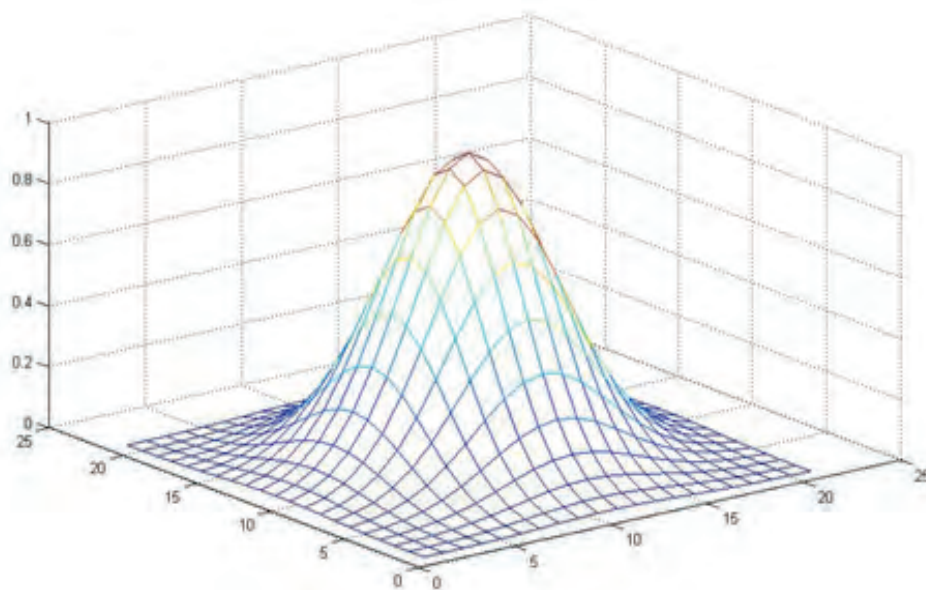


Figura 1: La función de pico único es fácil para los métodos basados en cálculo.

Primero, ambos métodos son de alcance local; los óptimos que buscan son los mejores en un vecindario del punto actual. Por ejemplo, suponga que la figura 1 muestra una parte del dominio de interés; una imagen más completa se muestra en la figura 2. Claramente, comenzar la búsqueda o los procedimientos de búsqueda en la vecindad del pico más bajo hará que perdamos nuestro punto de interés (el pico más alto).

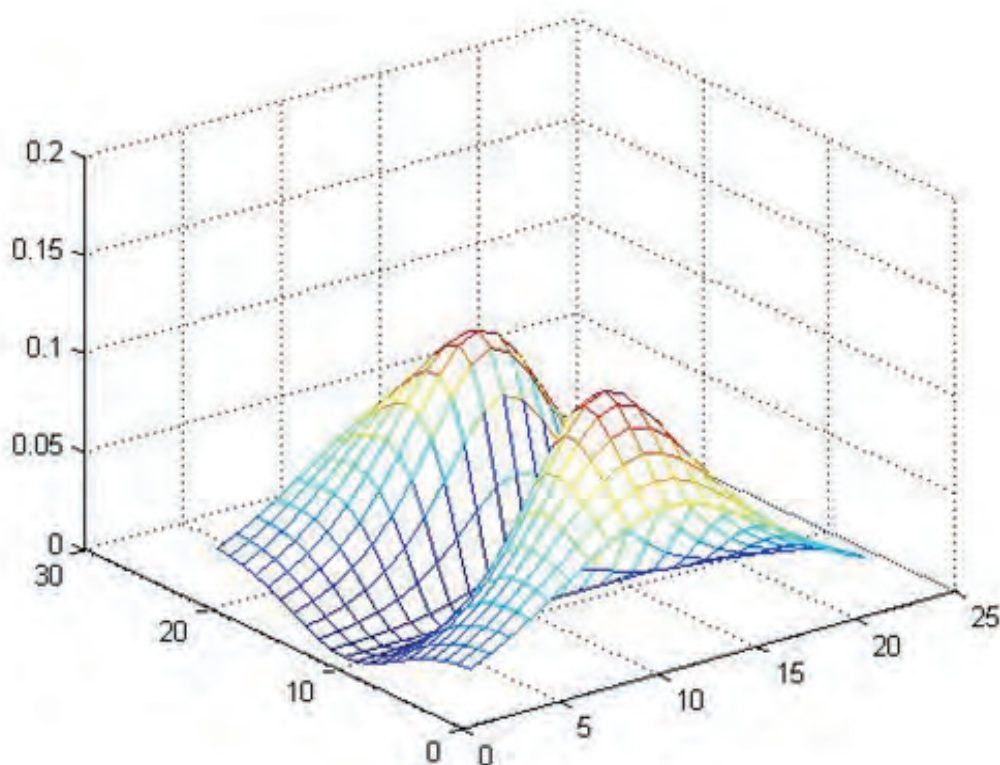


Figura 2: La función de picos múltiples plantea un dilema. ¿Desde qué punto deberíamos empezar la búsqueda?

Una vez que se alcanza el pico más bajo, se deben buscar mejoras adicionales mediante reinicios aleatorios u otras herramientas. Por otra parte, métodos basados en cálculo dependen de la existencia de derivadas (valores de pendiente bien definidos). Incluso si calculamos una aproximación numérica de las derivadas, esto representa en algunos casos un desacierto.

Por lo tanto, para aquellos interesados en la optimización deben aceptar solo parcialmente el legado de los grandes matemáticos de los siglos XVIII y XIX que pintaron un mundo limpio de funciones objetivas cuadráticas, restricciones ideales y derivadas siempre presentes, porque, el mundo real de la búsqueda está plagado de discontinuidades y vastos espacios de búsqueda multimodales y ruidosos (*Goldberg, 1989*), como se muestra en una función menos amigable con el cálculo en la figura 3. Por lo tanto, no sorprende que los métodos que dependen de los requisitos restrictivos de continuidad y existencia de la derivada son inadecuados todos los casos, excepto para un dominio de problemas muy limitado. Por esta razón y debido a su alcance de búsqueda inherentemente local, debemos rechazar los métodos basados en el cálculo. Debido a que no son lo suficientemente robustos en dominios no deseados (*Vanegas & Barragán, 2011*).

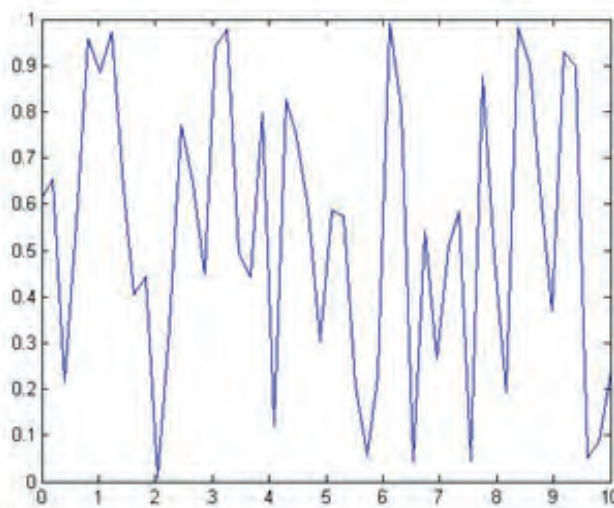


Figura 3: Muchas funciones son ruidosas o discontinuas y, por tanto, inadecuadas para la búsqueda por métodos tradicionales.

Esquemas de búsqueda enumerativos: Lo primero es mencionar que existe gran diversidad de métodos de búsqueda enumerativos en cuanto formas y tamaños se refiere. La idea general es bastante sencilla: dentro de un espacio de búsqueda finito, o un espacio de búsqueda infinito discreto, el algoritmo de búsqueda comienza a mirar los valores de la función objetivo en cada punto del espacio, uno a la vez (*Nebro & Luna, 2006*). Aunque la sencillez de este tipo algoritmo es atractiva y la enumeración es un tipo de búsqueda muy humano (cuando el número de posibilidades es pequeño), estos esquemas deben descartarse en última instancia en la carrera por la robustez por una simple razón: falta de eficiencia, cuando los espacios son demasiado grandes para buscar uno a la vez.

Los algoritmos de búsqueda aleatoria han alcanzado una popularidad creciente a medida que los investigadores han reconocido las deficiencias de los esquemas enumerativos y los basados en cálculo. Sin embargo, búsquedas aleatorias y esquemas aleatorios que buscan y guardan lo mejor también presentan desventajas debido al requisito de eficiencia. No se puede esperar que las búsquedas aleatorias, a largo plazo, funcionen mejor que los esquemas enumerativos (*D. Rios, 2000*). El algoritmo genético es un ejemplo de un procedimiento de búsqueda que utiliza la elección aleatoria como herramienta para guiar una búsqueda altamente exploradora a través de la codificación de un espacio de parámetros. Utilizar la elección aleatoria como herramienta

en un proceso de búsqueda dirigida parece extraño al principio, pero la naturaleza contiene muchos ejemplos. En *(Davis, 1987)* se explora las conexiones entre el recorrido simulado y los algoritmos genéticos. Lo importante de reconocer esta similitud, es que la búsqueda aleatoria no implica necesariamente una búsqueda sin dirección.

Para tener una mejor perspectiva, del breve contraste realizado en los anteriores párrafos observemos el espectro de problemas de la **figura 4**. En la figura se muestra un índice de eficiencia al momento de resolver un tipo de problemas con un esquema especializado, esquema enumerativo y esquema robusto idealizado. La técnica de gradiente funciona bien en un grupo limitado de problemas, como esperamos, pero se vuelve altamente ineficiente en otros lugares. Por otro lado, el esquema enumerativo funciona con ineficiencia igualitaria en todo el espectro de problemas, como lo muestra la curva de rendimiento inferior. El ideal es diseñar un método de búsqueda como el que aparece en la curva de rendimiento denominada esquema robusto. Valdría la pena sacrificar el rendimiento máximo en un problema particular para lograr un alto nivel de rendimiento en todo el espectro de problemas. (Por supuesto, con métodos amplios y eficientes siempre podemos crear esquemas híbridos que combinen lo mejor del método de búsqueda local con el esquema robusto más general.), Por lo cual los algoritmos genéticos ayudan a llenar esta brecha de robustez *(Goldberg, 1989)*.

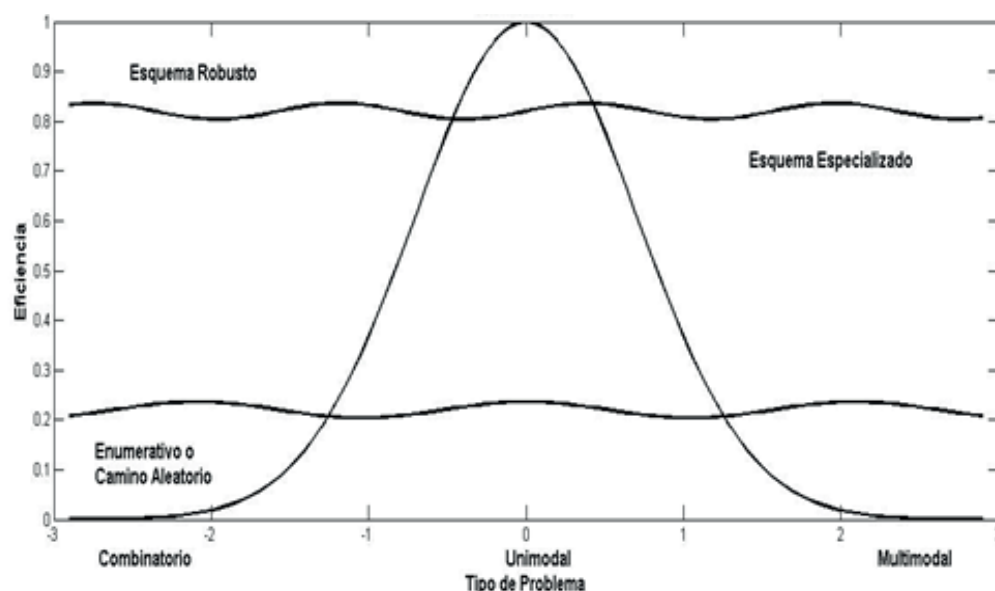


Figura 4: Muchos esquemas tradicionales funcionan bien en un ámbito de problemas limitado. Los esquemas enumerativos y los recorridos aleatorios funcionan igualmente ineficazmente en un amplio espectro. Un método sólido funciona bien en un amplio espectro de problemas.

Vocabulario relacionado con los algoritmos genéticos:

Los algoritmos genéticos utilizan un vocabulario tomado de la genética natural. Habitualmente se habla de individuos en una población; bastante a menudo estos individuos se denominan también cadenas de cromosomas. Esto podría ser un poco confuso: cada célula de cada organismo de una especie determinada lleva un cierto número de cromosomas el hombre, por ejemplo, tiene 46 de ellos. Los cromosomas están a su vez formados por unidades o genes (también llamados características, caracteres o decodificadores) arreglados en sucesión lineal, cada gen controla la herencia de uno o varios caracteres. Se localizan genes de ciertos caracteres en ciertos lugares del cromosoma, que se denominan posiciones de las cuerdas. Cualquier carácter de las personas (como el color del cabello) puede manifestarse de manera diferente.

Se dice que el gen está en varios estados, llamados alelos (valores característicos). Cada genotipo, representaría una solución potencial a un problema (el significado de un cromosoma particular, es decir, su fenotipo, es definido externamente por el usuario) (Turnpenny, 2009), un proceso de evolución ejecutado en una población de cromosomas corresponde a una búsqueda a través de un espacio de soluciones potenciales. Tal búsqueda requiere equilibrar dos objetivos: aprovechar las mejores soluciones y explorar el espacio de búsqueda.

Aspectos que se deben tener en cuenta para la implementación de un Algoritmo genético
Antes de mencionar los aspectos para la implementación de un algoritmo genético, en cualquier lenguaje de programación observemos el siguiente diagrama, que representa los procesos que se van a desarrollar.



Figura 5: Diagrama de procesos en un algoritmo genético.

PASO 1: Como ya se ha mencionado los algoritmos genéticos, son algoritmos que permiten optimizar, por lo tanto, es muy importante definir una función objetivo.

PASO 2: Los algoritmos genéticos procesan cadenas de poblaciones. Por lo tanto, la estructura de datos principal para un algoritmo genético simple es una población. Hay varias formas de implementar poblaciones. Para el algoritmo genético una de las formas básicas, que se puede elegir, sería: construir una población como una matriz de individuos donde cada individuo contiene el fenotipo (el parámetro decodificado o parámetros), el genotipo (el cromosoma artificial o cadena de bits).

PASO 3: La declaración de una serie de constantes, el tamaño máximo de población, y la longitud máxima de la cadena. Con el fin de establecer límites superiores en el tamaño de la población y la longitud de la cadena. Es de gran importancia declarar esta información de esta manera caracterizamos la población en sí, junto con sus componentes. El tipo de individuo corresponde al registro del tipo de cromosoma.

PASO 4: La aplicación de operadores genéticos a una población completa en cada generación, como se muestra en la figura 6. Para implementar esta operación limpiamente, utilizamos dos poblaciones no superpuestas, lo que simplifica el nacimiento de la descendencia y el reemplazo de los padres. Con estas dos poblaciones, es muy sencillo crear una nueva descendencia a partir de los miembros utilizando los operadores genéticos, colocamos esos nuevos individuos en una función asignada para ello y establecemos la segunda población como la nueva población de partida. Cabe aclarar que existen otros métodos de manipulación de poblaciones

más eficientes en el almacenamiento. Podríamos mantener una sola población superpuesta y prestar más atención a quién reemplaza a quién en poblaciones sucesivas. Tampoco hay ninguna razón particular para mantener constante el tamaño de la población. Las poblaciones naturales ciertamente cambian en tamaño, y puede haber motivación durante la búsqueda genética artificial para permitir la variación del tamaño de la población de una generación a otra. Sin embargo, por el momento se mantendrán las cosas simples, y esto será la elección de poblaciones no superpuestas de tamaño constante. Una vez se tienen las estructuras de datos diseñadas y construidas, necesitamos comprender los tres operadores: selección, cruce y mutación, esenciales para la operación de algoritmos genéticos.

Selección: Consiste en escoger un individuo o varios individuos (según sea el caso) de manera aleatoria de entre los individuos más aptos, a fin de prolongar su material genético en las siguientes generaciones (Goldberg, 1989).

Cruce: Posterior a la selección se realiza el cruce, durante este procedimiento se generan nuevos individuos a partir de la información genética de sus padres, quienes formarán parte de una nueva población. Esta operación se repetirá hasta que el número definido de habitantes sea alcanzado (Turnpenny, 2009).

Mutación: Cada individuo nuevo se someterá a la mutación, en la cual se modifica su información genética. Generalmente, la mutación consiste en la inversión del valor de uno de sus cromosomas (Turnpenny, 2009). Los cambios generados en la información genética del individuo permiten al algoritmo realizar cambios en la región de búsqueda, posibilitando el salto hacia un máximo global en caso de un estancamiento ocasionado por la proliferación de individuos similares (Goldberg, 1989).

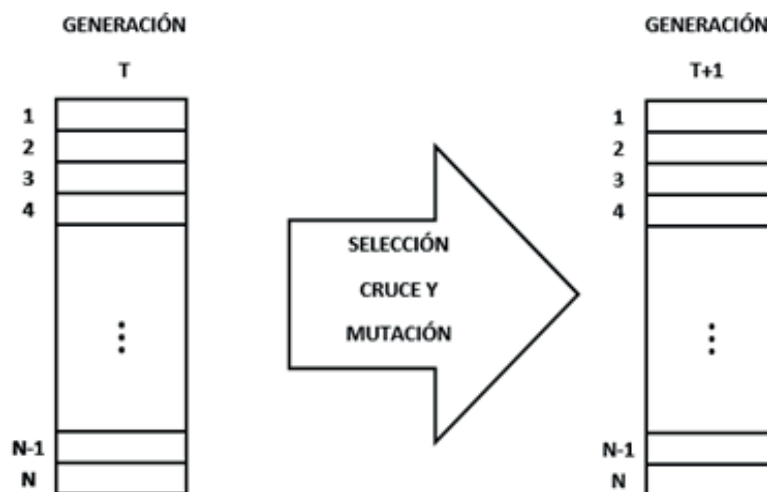


Figura 6: Esquema de poblaciones no superpuestas en el algoritmo genético.

PASO 5: Establecer un criterio de parada que finaliza el proceso generacional. Dos criterios comunes son: número de generaciones, es decir, una vez se alcanza un de generaciones establecido finaliza la búsqueda. Otro criterio consiste en verificar la información genética de la población, en la cual detiene la evolución cuando se verifica que hay una proliferación de un mismo individuo. Al finalizar el algoritmo, se escoge el mejor habitante de la población final y se espera que un material genético proporcione la mejor respuesta al problema (Díaz, 2012).

Ejemplo 2:

A continuación, se presenta un ejemplo muy sencillo, empleado para estimar parámetros de una ecuación diferencial no lineal. La búsqueda se realiza mediante un sencillo código programado en el software Matlab y se contrasta con datos obtenidos de manera experimental. La ecuación que describe la dinámica del fenómeno estudiado es la siguiente:

$$\frac{dx}{dt} = \alpha x \left(1 - \frac{x}{k}\right).$$

Los parámetros a estimar son α y k , los cuales corresponden a tasa de crecimiento y capacidad de carga. Es ecuación anterior se conoce con el nombre de ecuación logística, ha sido ampliamente estudiada y aplicada para modelar fenómenos en diferentes áreas del conocimiento (Gómez Hernández & Ibargüen-Mondragón, 2018).

```
1 A=xlsread('datosecoli.xlsx');
2 time=A(:,1);
3 t=linspace(0,time(end),81)';
4 resist=A(:,3);
```

Dado que nuestro ejemplo, como ya se mencionó, busca estimar parámetros el primer paso es llamar el archivo en el cual tenemos almacenados los datos experimentales (**línea 1**). En las líneas 2 y 4 creamos dos vectores columna tomando los datos del archivo “datosecoli.xlsx” de sus columnas 1 y 3 las cuales corresponden al tiempo de observación y datos observados en dicho periodo de tiempo, finalmente en la línea 3 creamos un vector, igualmente espaciado desde 0 hasta el tiempo final de observación.

Se creó un archivo auxiliar, denominado infection en él se introdujeron los datos experimentales tomados con sus respectivos tiempos, como vectores, ya que posteriormente se utilizó la función de optimización de Matlab (línea 5), para comparar los datos de infection con los posibles parámetros. En la línea 6 se almacenan los parámetros iniciales (población inicial para empezar la búsqueda), estos se introducen en relación al fenómeno o pueden introducirse de manera totalmente aleatoria. Para acotar los valores de búsqueda (individuos de las generaciones siguientes) en las líneas 7 y 8 se almacenan, α_R y k_R , empezando la búsqueda en $\text{Parametros}(1)*0.25$ y finalizando la misma en $\text{Parametros}(1)*2$, es decir, para el caso de α inicia en $0.282*0.25$ y finaliza en $0.282*2$.

```
9 alphaP=alphaR(1)+rand*(alphaR(2)-alphaR(1));
10 kP=kR(1)+rand*(kR(2)-kR(1));
11 nt=200;
```

En las siguientes líneas (9 y 10) se almacenan los individuos que formaran la población inicial, para no sesgar la selección se hace uso de la función rand que corresponde a una lista de números aleatorios proporcionados desde el software y en la línea 11 se almacena un número denominado

nt, posteriormente utilizaremos este valor en el uso de la función de algoritmo genético.

```
12 opts=gaoptimset('EliteCount',0.1*nt,'Generations',100,'PopulationSize',nt,'InitialPopulation',[
    alphaR kP],'CrossoverFraction',0.8,'MutationFcn',@mutationadaptfeasible,'PopInitRange',
    [alphaR(1) kR(1); alphaR(2) kR(2)],'CreationFcn',@gacreationuniform,'SelectionFcn',
    @selectionstochunif);
```

En la **línea 12** utilizamos una de las opciones de optimización, la cual devuelve una lista de opciones y los valores predeterminados para el algoritmo seleccionado que para nuestro ejemplo es gaoptimset. La lista de opciones incluye EliteCount corresponde a un entero positivo que especifica cuántos individuos de la generación actual están garantizados para sobrevivir en la siguiente generación, Generations se constituye en un entero que indica el número de generaciones máximo a obtener, PopulationSize Tamaño de la población, InitialPopulation Población inicial utilizada para sembrar el algoritmo genético, CrossoverFraction La fracción de la población en la próxima generación, sin incluir a los niños de élite, que crea la función de cruce, MutationFcn Función que produce hijos de mutación. Especifique como nombre de una función de mutación incorporada o un identificador de función, PopInitRange Los rangos de valores para los individuos de la población inicial, CreationFcn Función que crea la población inicial. Especifique como nombre de una función de creación incorporada o un identificador de función, SelectionFcn Función que selecciona padres de hijos cruzados y mutantes. Especifique como nombre de una función de selección incorporada o un identificador de función.

```
12 [B,fval,output,population,scores]=ga(opt_fcn,2,[],[],[],[],[alphaR(1)
    kR(1)],[alphaR(2) kR(2)],[],opts)
13 [timefit,yfit]=ode23(@(t,x)resonant(t,x,B),t,[7.477121255]);
14 h=figure();
15 plot(time,resist,'pg','MarkerFaceColor','r','markersize',12)
16 hold on
17 plot(timefit,yfit(:,1),'LineWidth',3)
```

En la **línea 13** programamos los datos que deseamos que imprima el software una vez ha realizado la búsqueda de parámetros, entre ellos están el error, la población seleccionada, las generaciones obtenidas en total. La línea 14 corresponde a la solución de la ecuación diferencial, empleando la opción ode23. Finalmente, las líneas 15 a la 18 permiten observar las gráficas, esto es opcional.

Finalmente aparece un listado con las poblaciones de cada una de las generaciones encontradas, el error estimado para los parámetros y si el usuario lo desea la gráfica.

0.2814	8.6119
0.2813	8.6120
0.2813	8.6120
0.2813	8.6117
0.2813	8.6119
0.2812	8.6118
0.2813	8.6118
0.2812	8.6118

Figura 7: Últimas 8 generaciones estimadas para α (primer columna) y k (segunda columna).

En la siguiente grafica observamos la simulación ejecutada (línea continua) mediante el uso del software, dicha simulación se realizó con los valores de los parámetros estimados,

adicionalmente se incluye en esta figura los datos obtenidos experimentalmente en laboratorio (estrellas), con el objetivo de contrastar ambos resultados.

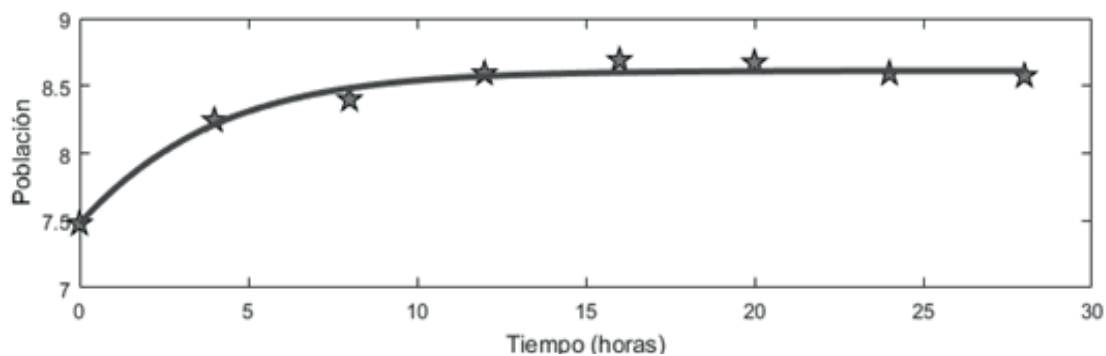


Figura 8: Grafica obtenida al estimar parámetros, en la cual se observa la curva estimada y los puntos experimentales

APLICACIONES DE LOS ALGORITMOS GENÉTICOS

Los algoritmos genéticos se han aplicado con bastante éxito a problemas de optimización, como enrutamiento de cables, programación, control adaptativo, juego de ganancia, modelado cognitivo, problemas de transporte, problemas del viajante de comercio, problema de control óptimo, optimización de consultas a la base de datos, problemas de ingeniería y afines y problemas de estimación de parámetros en problemas aplicados a la biología, siendo estos últimos de interés particular en este documento.

Aplicación a la microbiología:

La resistencia bacteriana es un tema de interés debido a su creciente presencia motivo por el cual esta constituye un problema de interés científico. Diversos estudios se han realizado alrededor de este tema, fijando su propósito en conocer mejor este fenómeno y así prevenir y controlar el mismo (Ibarguen Mondragon, y otros, 2014). Para ello se han formulado diferentes modelos poblacionales, a continuación, se presenta un modelo de cinética poblacional, dicho modelo fue contrastado con datos obtenidos (población de *E. Colli*) experimentalmente a partir de muestras hospitalarias aisladas, posteriormente se estimaron los parámetros del modelo con fines predictivos. La dinámica poblacional se resume en el siguiente diagrama:

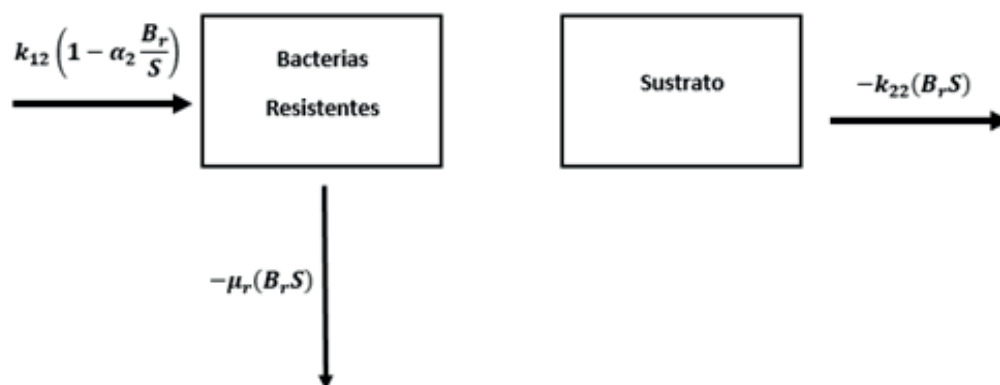


Figura 9: Diagrama de flujo que describe la dinámica poblacional de un cultivo de *E. Colli*

aislado.

El sistema de ecuaciones diferenciales, que describe la dinámica poblacional se presenta a continuación:

$$\frac{dB_r}{dt} = k_{12} \left(1 - \alpha_2 \frac{B_r}{S} \right) B_r - \mu_r B_r$$

$$\frac{dS}{dt} = -k_{22} B_r S$$

Se formuló un modelo tipo logístico, con capacidad de carga variable (*Ibarguen Mondragon, Romero Leiton, Esteva, Ceron Gomez, & Hidalgo Bonilla, 2010*) en correspondencia a la limitación de sustrato que se presenta en el medio de cultivo, este describe la interacción de E. coli resistente-sustrato, las variables que intervienen son B_r que representa la población de E.coli resistente y S que corresponde al concentración de sustrato, los parámetros propuestos para este modelo son: k_{12} , tasa de crecimiento poblacional, α_2 parámetro de ajuste, μ_r tasa de muerte natural de E. coli resistente y k_{22} tasa de degradación de sustrato. Los parámetros a estimar son: k_{12} , α_2 y k_{22} . El parámetro μ_r se fijó en 1/72 de acuerdo con la bibliografía revisada.

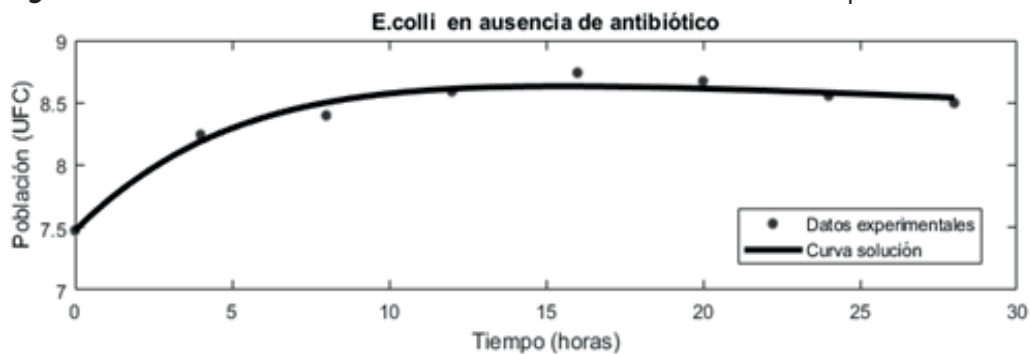
La solución numérica se realizó mediante la implementación y programación de un algoritmo genético en el software Matlab. Se estableció la función objetivo, solucionando el sistema de ecuaciones diferenciales empleando la función ode23 y contrastando dicha solución con los datos experimentales. La población inicial se acota a fin de economizar recursos y teniendo en cuenta la información estudiada sobre el fenómeno en cuestión. Con la función de optimización de algoritmos genéticos incorporada en el software se introdujeron los aspectos propios de este algoritmo (población inicial, tamaño de la población, numero de generaciones), adicionalmente las operaciones de selección, mutación y cruce.

Los resultados obtenidos sintetizan en las siguientes tabla y gráfica:

Tabla 1: Valores obtenidos en la estimación.

PARAMETRO	ESTIMACIÓN	UNIDADES
K_{12}	0,2342	$1/h$
K_{22}	0,0001	$1/(UFC * mL)$
α_2	10,6921	mL/UFC

Una vez estimados los parámetros mediante el algoritmo genético, se procedió a realizar una simulación de la curva con dichos parámetros, en la figura 10 se observa la curva simulada en contraste con los valores obtenidos de manera experimental con el material de E. coli hospitalario aislado.

Figura 10: Gráfico de solución en contraste con los datos experimentales

Aplicación a la epidemiología:

La epidemiología enfrenta variados problemas (Gomez Hernandez, Gonzalez Santacruz, & Ibarguen Mondragon, 2014), entre los cuales encontramos el alto índice de contagio de VIH (virus de la inmunodeficiencia humana) siendo África y Latinoamérica los continentes con mayor presencia tanto de VIH como de SIDA. En (Jorge Barrios Ginart, 2010) se formula un modelo estocástico de ecuaciones diferenciales, el cual describe la epidemia de VIH/SIDA en Cuba. Posteriormente se estiman los parámetros, con el fin de realizar la interpretación y validación del modelo. Aunque han sido varios los modelos formulados para describir la dinámica de VIH/SIDA en Cuba, en Barrios et. al. (2010) se consideran las siguientes variables.

$X(t)$: Representa el número de personas infectadas con VIH que no conocen que están infectados en el momento t .

$Y_1(t)$: Representa el número de personas infectadas con VIH que conocen que lo están en el momento t , y que fueron detectados aleatoriamente (análisis de rutina, prueba a embarazadas, etc.).

$Y_2(t)$: Representa el número de personas infectadas con VIH que conocen que lo están en el momento t y que se descubrieron a través de la búsqueda por contactos.

$Z(t)$: Representa el número de personas enfermas con SIDA en el momento t .

La dinámica se describe en el siguiente sistema de EDOs

$$\frac{\partial X}{\partial t} = \lambda X + \lambda'(Y_1 + Y_2) - (k_1 + \mu + \beta)X - q(k_2, X, Y_1, Y_2)$$

$$\frac{\partial Y_1}{\partial t} = k_1 X - (\mu + \beta')Y_1$$

$$\frac{\partial Y_2}{\partial t} = q(k_2, X, Y_1, Y_2) - (\mu + \beta')Y_2$$

$$\frac{\partial Z}{\partial t} = \beta X + \beta'(Y_1 + Y_2) - \mu'Z$$

Tabla 2: Descripción de los parámetros empleados en el sistema de ecuaciones.

NOTACIÓN	PARÁMETRO
λ	Tasa de contagios causados por individuos cero positivos no detectados.
λ'	Tasa de contagios causados por individuos cero positivos detectados
k_1	Tasa con que una persona desconocida e infectada con VIH es detectada por el sistema de forma aleatoria.
k_2	Tasa con que una persona desconocida e infectada con VIH es detectada por el sistema a través de la búsqueda de contactos.
β	Tasa con que personas infectadas con VIH, no detectadas, desarrollan el SIDA.
β'	Tasa con que personas infectadas con VIH, detectadas, desarrollan el SIDA (el recíproco del tiempo que va entre Y y Z).
μ	Tasa de mortalidad de la población sexualmente activa.
μ'	Tasa de mortalidad de la población con SIDA.

En esta investigación y a partir de aspectos considerados por los investigadores, tales como encuestas y estudios previos, se acotaron algunos de los parámetros restringiendo la búsqueda y economizando recursos de máquina y software.

La solución numérica se realizó, teniendo en cuenta la siguiente estrategia:
Encontrar la solución del sistema de EDOs.

Encontrar el valor del vector de parámetros $p=(\lambda, \rho, k_1, k_2)$ que minimice la función cuando estos tienen restricciones de cotas.

La estimación de parámetros se realizó en el software MATLAB. Para la integración numérica de los sistemas de EDOs se utilizaron las funciones de MATLAB ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb y se programaron las funciones LL y LLRK4 que implementan los métodos de linealización Local y linealización Local-Runge Kutta de orden 4 respectivamente.

Para la minimización de la función con restricciones de cotas se utilizaron las funciones fmincon y lsqnonlin del toolbox de optimización de MATLAB que implementan métodos deterministas y la función ga del toolbox Algoritmo Genético y búsqueda directa de MATLAB que implementa un Algoritmo Genético, elegir el algoritmo genético como método de optimización, combinado con cualquier integrador de los mencionados constituye la vía no determinista de la estrategia de solución.

Aplicación a la ecología:

La formulación de modelos en sistemas ecológicos resulta de gran utilidad, sin embargo, un

gran problema para el desarrollo y aplicación práctica de estos modelos es la dificultad de obtener valores adecuados para los parámetros debido a la enorme complejidad de los sistemas ecológicos y las grandes dificultades operativas para obtención de datos (*Ebecken, 2000*).

Entonces aparece la calibración como una herramienta importante para refinar estimaciones disponibles y proporcionar cifras adecuadas para los parámetros en los que no fue posible obtener información confiable. Aunque la calibración es básicamente un problema de optimización, características como no linealidad y funciones de costos multimodales, son comunes en los modelos ecológicos, esto representa un obstáculo en el momento de emplear métodos tradicionales de búsqueda y optimización. Como consecuencia, la calibración se realiza normalmente con un manual, con gran consumo de tiempo y con resultados probablemente inferiores a los que podría obtenerse mediante un método automático. En (*Ebecken, 2000*) desarrollaron una nueva herramienta para la calibración automática, basada en algoritmos genéticos (AG). El AG busca el mejor conjunto de valores, dentro de un espacio de búsqueda determinado usando la función de costo, el error cuadrático medio, calculado a partir de la solución numérica del EDOs, para cada combinación de parámetros generada por el AG. El programa de calibración automática desarrollado fue probado usando el modelo predador-presa Hassel-Varley, con 7 parámetros que deben proporcionarse. La calibración fue realizada en relación con datos artificiales, obtenidos por la resolución numérica de EDOs para un conjunto de parámetros conocidos, agregando luego un término estocástico, con distribución gaussiana.

CONCLUSIONES

A partir de la literatura se observa que los métodos de búsqueda tradicionales presentan limitaciones que restringen su utilidad y aplicabilidad a cierto tipo de problemas.

Los esquemas mencionados en este documento han sido utilizados con éxito en diversas aplicaciones, sin embargo, a medida que se abordan problemas de mayor complejidad que no se ajusten a las hipótesis del método es necesario emplear otros métodos.

Debido a que los algoritmos genéticos son métodos de búsqueda aleatorios enfocados en la optimización, la gama de problemas en los cuales se pueden aplicar es amplia y variada, esto permite evidenciar la versatilidad y adaptabilidad de este método.

En cuanto a las aplicaciones a la biología (microbiología, epidemiología, ecología, etc.) los algoritmos genéticos se emplean con frecuencia en la estimación de parámetros de modelos predictivos, permitiendo a los investigadores aproximarse de manera confiable a la comprensión del fenómeno biológico investigado y por tanto poder tomar decisiones acertadas.

Cabe resaltar que el algoritmo genético presenta relativa sencillez en cuanto al diseño e implementación en un software una vez se ha identificado el problema. Como desventaja se puede mencionar que el consumo de recursos (tiempo, hardware y software) podrían ser más o menos, en relación a la complejidad del problema a resolver.

REFERENCIAS BIBLIOGRÁFICAS

D. Rios, S. R. (2000). Simulación: Métodos y aplicaciones. Bogota: Alfaomega.

Davis, F. (Agosto de 1987). User acceptance of information system: the technology acceptance model . User acceptance of information system: the technology acceptance model . Michigan, Estados Unidos.

- Diaz, E. Y. (2012). Aplicación de algoritmos genéticos en problemas. Investigación innovación ingeniería, 10-29.
- Ebecken, C. B. (2000). Calibração Automática de um Modelo Predador-Presa. Brasil.
- Goldberg, D. E. (1989). ALGORITHMS GENETICS insearch, Optimization and Machine Learning. New York: Addison-Wesley Publishing company, Inc.
- Gomez Hernandez, E. A., Gonzalez Santacruz, E. C., & Ibarguen Mondragon, E. (2014). Elementos para la modelación matemática de la epidemiología de la neumonía. Universidad y salud, 219-227.
- Gomez Hernandez, E. A, & Ibarguen Mondragon, E. (2018)- Modeling the Dynamics of the Mosquito. Contemporary Engineering Sciences, 871 - 879
- Ibarguen Mondragon, E., Mosquera, S., Ceron, M., Burbano, E. M., Hidalgo Bonilla, S. P., Esteva, L., & Romero Leiton, J. P. (2014). Mathematical modeling on bacterial resistance to multiple antibiotics caused by spontaneous mutations. BioSystems, 60-67.
- Ibarguen Mondragon, E., Romero Leiton, J. P., Esteva, L., Ceron Gomez, M., & Hidalgo Bonilla, S. P. (2010). Stability and periodic solutions for a model of bacterial resistance to antibiotics caused by mutations and plasmids. Applied mathematical modelling, 238-251.
- Jorge Barrios Ginart, A. d. (2010). Estimación de parámetros en modelos epidemiológicos de VIH/SIDA. Revista de Matemática: Teoría y Aplicaciones, 143-158.
- Jurado, J. M. (1 de Abril de 2008). Aplicación de Microsoft Excel a la Química Analítica: validación de métodos analíticos. Aplicación de Microsoft Excel a la Química Analítica: validación de métodos analíticos. Sevilla, España.
- Macias, G. b. (1973). Comparacion de dos métodos de busqueda lineal en problemas de optimización. Monterrey, Mexico.
- Michalewicz, Z. (1999). Gnetic algorithms+Data structures=Evolution programs. Charlotte: Springer.
- Nebro, A. J., & Luna, E. A. (2006). Multi-Objective Optimization using Grid Computing. Springer, 50-62.
- Turnpenny, P. D. (2009). Elementos de genética medica. Barcelona: Elsevier.
- Vanegas, D., & Barragán, K. S. (Enero-Junio de 2011). Real potencial del uso del método de análisis de intervalos para la optimización con restricciones frente al PSO de convergencia garantizada. Comparación de las técnicas de optimización por análisis de intervalos y la de enjambre de partículas para funciones con restricciones. Bogota, Colombia.